



TAMPEREEN TEKNILLINEN YLIOPISTO

MIKKO AIMO

**MICROSOFT DYNAMICS AX – QLIKVIEW
-OHJELMISTOKEHITYSPROSESSI**

Diplomityö

Tarkastaja: Prof. Tommi Mikkonen
Tarkastaja ja aihe hyväksytty Tieto- ja
sähkötekniikan tiedekuntaneuvoston
kokouksessa 7. lokakuuta 2009

TIIVISTELMÄ

Tampereen Teknillinen Yliopisto

Tietotekniikan koulutusohjelma

AIMO, MIKKO: Microsoft Dynamics AX – QlikView -ohjelmistokehitysprosessi

Diplomityö, 51 sivua

Helmikuu 2015

Pääaine: Ohjelmistotuotanto

Työn tarkastaja: Professori Tommi Mikkonen

Avainsanat: Liiketoimintatiedon hallinta, QlikView, Microsoft Dynamics AX, ETL, ERP, Scrum, Lean

Informaatioteknologia tuottaa jatkuvasti uusia ohjelmistoja. Yritysten haasteena on pitää ohjelmistonsa ajan tasalla vaihtamatta järjestelmiään liian usein. IT-ohjelmistoja myyvät tahot auttavat yrityksiä arvioimaan uusien ohjelmistoratkaisujen tarvetta. Uuden tuotteen ottaminen IT-ohjelmistoja jälleen myyvän yrityksen tuoteportfolioon on pitkäkestoinen prosessi, ja sen onnistumista voidaan arvioida vasta vuosien päästä.

Tässä diplomityössä arvioitiin QlikView-ohjelmiston soveltuvuutta SYSteam Business Solutions Oy:n tuoteportfolioon. Analysointi toteutettiin projektilla, jossa QlikView-ohjelmistolla toteutettiin kahden kuukauden aikana kahdeksan Microsoft Dynamics AX toiminnanohjausjärjestelmään perustuvaa roolipohjaista raporttia. Raportit suunniteltiin uudelleen käytettäväksi ja nopealla räätälöinnillä käyttöönotettaviksi.

Projektin tavoitteena oli todeta QlikView-tuotteen sopivuus SYSteamin tuotevalikoimaan, kartuttaa QlikView-ohjelmisto-osaamista ja löytää kehittäjien joukosta uusi vakituinen työntekijä Business Intelligence -kehitystiimiin.

Projektin alussa raporttien kehittäjät koulutettiin sekä QlikView- että Dynamics AX -järjestelmien käyttöön ja kehittämiseen. Toteutusvaiheessa kehittäjillä oli kuukausi aikaa toteuttaa määrittelyjen mukaiset raportit. Projektin loppuvaiheessa raporteista tehtiin dokumentaatiot. Projektin päättyessä kaikki projektiin osallistuneet haastateltiin ja projektin tuotoksia sekä tuloksia arvioitiin.

Kaikki raportit yhtä lukuun ottamatta saatiin määrääkaaan mennessä valmiiksi. Toteutetuista raporteista kolmea saatiin myytyä asiakkaille. Kaikkia raportteja pystyttiin käyttämään hyödyksi muissa QlikView-kehitysprojekteissa.

Projektin tavoitteet saavutettiin. Sekä asiakastyytyväisyys että QlikView-ohjelmiston kehittäjätyytyväisyys puolsivat sitä, että QlikView valittiin SYSteamin tarjoamaan. Projektilla saatiin hyvä pohja QlikView-osaamiselle ja projektin seurauksena SYSteamille palkattiin vakituinen työntekijä BI-kehitystiimiin.

ABSTRACT

Tampere University of Technology

Master's Degree Programme in Information Technology

AIMO, MIKKO: Microsoft Dynamics AX – QlikView –Software Development Process

Master of Science Thesis, 51 pages

February 2015

Major: Software Engineering

Examiner: Professor Tommi Mikkonen

Key words: Business Intelligence, QlikView, Microsoft Dynamics AX, ETL, ERP, Scrum, Lean

New software is continuously produced by information technology. Thus, it is a challenge for any company to maintain the relevancy of their software. Software selling companies help other companies to assess the need for new software solutions. Launching a new software product is a long process for the software selling company and the success of the launch can only be evaluated after years of experience.

The focus of this master's thesis was to evaluate the suitability of QlikView software for the product portfolio of SYSteam Business Solutions Oy. The evaluation was carried out by a project where eight reports were made by using QlikView software. The reports were role based and originated from Microsoft Dynamics AX enterprise resource planning system. The reports were designed to be modular and quickly tailorable for the customer.

The goal of the project was to decide whether or not to accept QlikView as a new product for SYSteam. Another goal was to increase the knowledge and skills for the QlikView software and to find another suitable employee for the Business Intelligence – team.

In the beginning of the project all developers were trained to use and develop both QlikView and Dynamics AX software. The developers had one month to finish the reports. At the end of the project all the participants were interviewed and the results were analyzed.

All reports except one were finished within the given deadline. Three of the reports were successfully sold to customers. However, all of the reports could be used to benefit other QlikView projects.

The goals of the project were achieved. QlikView was found to be a developer-friendly software and the customers were satisfied to the extent that QlikView was chosen as a new product for SYSteam's product line. The knowledge and skills of the developers were greatly increased during the project and at the end of the project a suitable candidate was hired.

ALKUSANAT

Tämä diplomityö on tehty osana SYSteam Business Solutions Oy:n, jäljempänä SYSteam, ohjelmistokehitysprojektia Tampereella vuosina 2009 ja 2010. Työssä arvioidaan SYSteamin ohjelmistokehitysprojektia, sen liiketoiminnallista hyötyä ja esitellään mahdollisia parannusehdotuksia ohjelmistokehitysprojektin kehittämiseksi. Työ on tehty SYSteamin uusia ja tulevia ohjelmistoprojekteja varten. Työn tuloksilla on merkitystä tuleviin SYSteamin ohjelmistokehityshankkeisiin ja niiden läpivienteihin. Diplomityön tavoitteena on auttaa SYSteamin työntekijöitä ohjelmistokehityksessä, kehitysmallien valinnassa ja antaa näkökulma uuden ohjelmistokehitysmallin käyttöönottoon ja käyttöönoton ongelmiin.

Tampere, 8.1.2015

Mikko Aimo

SISÄLLYS

1.	Johdanto	1
2.	Liiketoimintatiedon hallinta	3
2.1.	Tieto	3
2.2.	Tiedon hankinta.....	4
2.3.	Työkalut	5
2.4.	Liiketoimintatiedon hallinnan tavoitteet	5
3.	Liiketoimintatiedon hallinta prosessina	7
3.1.	Tietokannat.....	7
3.2.	Tietovarastot.....	7
3.2.1.	Paikallisvarastot	8
3.2.2.	ETL	9
3.3.	Raportointi	10
3.4.	What-if	11
4.	Käytetyt Ohjelmistot	12
4.1.	QlikView-ohjelmisto.....	12
4.1.1.	Qlik	12
4.1.2.	QlikView.....	13
4.1.3.	QlikView-raportit.....	16
4.2.	Tietojärjestelmä.....	18
4.2.1.	ERP-järjestelmä	20
4.2.2.	Microsoft Dynamics Ax.....	21
5.	Ketterä ohjelmistokehitys.....	22
5.1.	Ohjelmistokehitysmalleja.....	22
5.1.1.	Lineaarinen vesiputous	22
5.1.2.	Scrum	23
5.1.3.	Lean	25
5.2.	Henkilöresurssien valinta	27

6.	Ketterä QlikView-kehitys	29
6.1.	Projektin eteneminen.....	29
6.1.1.	Reunaehdot	29
6.1.2.	Projektin toteutus	29
6.1.3.	Roolit	30
6.1.4.	Koulutukset.....	31
6.2.	Pyrähdykset.....	32
6.2.1.	Ensimmäinen pyrähdys.....	32
6.2.2.	Toinen pyrähdys	33
6.2.3.	Pyrähdys 2,5.....	34
6.2.4.	Kolmas pyrähdys	35
7.	Ratkaisun arviointi	36
7.1.	Haastattelut.....	36
7.1.1.	Haastattelukysymykset	36
7.2.	Arviointi	37
7.2.1.	Projektin kehittäjätyytyväisyys.....	37
7.2.2.	Raporttien asiakastytyväisyys.....	39
7.2.3.	Laatu	40
7.2.4.	Tehokkuus.....	40
7.2.5.	Hyödyt	40
7.3.	Kehitysideat.....	41
8.	Yhteenveto	42
	Lähteet.....	43

1. JOHDANTO

Informaatioteknologia tuottaa uusia ohjelmistoja nopealla tahdilla. Tähän vaikuttavat muun muassa tehokkaat ohjelmistojen kehitysmallit. Yritysten haasteena on pitää ohjelmistonsa ajan tasalla vaihtamatta tietojärjestelmää kuitenkaan liian aikaisin tai liian usein. Kaikki ohjelmistot käyvät jossain vaiheessa vanhaksi, jolloin on aika vaihtaa uuteen. IT-ohjelmistoja jälleen myyvät palvelutalot auttavat yrityksiä arvioimaan ja rakentamaan uusia ohjelmistoratkaisuja. Hyvän partnerin löytäminen jopa vuosia kestävään ohjelmistoprojektiin on erittäin tärkeää, ja sillä voi olla suuri merkitys projektin etenemiseen, tuloksiin ja projektista koituviin kustannuksiin.

Ohjelmistoja jälleen myyvät palvelutalot pyrkivät tavallisesti profiloimaan asiakaskuntaansa tai keskittymään tiettyyn asiakassegmenttiin, jotta tarvittava palvelun laatu tasoa pystytään takaamaan. Uuden tuotteen ottaminen yrityksen tuoteportfolioon on pitkäkestoinen prosessi, ja sen onnistumista voidaan arvioida vasta vuosien päästä.

Tässä diplomityössä tarkastellaan ja analysoidaan uuden tuotteen ottamista IT-palvelutalon tuoteportfolioon. Aluksi todennettiin ja varmennettiin tuotteen yhteensopivuus yrityksen muiden tuotteiden kanssa. Tämän jälkeen toteutettiin projekti, jossa oli määrä saada uusi tuote mahdollisimman helposti osaksi yrityksen tarjoamaa tuotevalikoimaa.

Diplomityössä analysoitu projekti, jossa uusi tuote otettiin yrityksen tuotevalikoimaan, on nimeltään 6pack. 6pack-projektissa toteutettiin kahdeksan Microsoft Dynamics AX -roolipohjaista QlikView-raporttia. Raportit auttavat SYSteamin asiakkaita heidän työtehtävissään antaen tietoa Dynamics AX -järjestelmän prosesseista. Raportin avulla työntekijä voi analysoida työtehtäviinsä liittyviä tietoja, etsiä tietoa päivittäiseen päätöksentekoonsa sekä nopeuttaa päätöksentekoprosessia.

6pack-raporttien kehitysprosessi ei noudattanut suoraan mitään tunnettua ohjelmistokehitysmallia. Projektia varten laadittiin oma kehitysmalli, jota on arvioitu tässä diplomityössä. Vaikka kehitysmalli ei ole yhtenäinen minkään nykyisen ohjelmistokehitysmallin kanssa, se sisältää monia hyväksi havaittuja ominaisuuksia tunnetuista ohjelmistokehitysmalleista.

6pack-projektiin osallistui kesällä 2009 yhdeksän SYSteamin työntekijää, joista kahdeksan opiskeli TTY:llä. Yhdeksästä projektiin osallistuneesta SYSteamiläisestä viisi palkattiin tässä työssä arvioitua 6pack-projektia varten. Nämä valitut olivat Tampereen Teknillisen Yliopiston opiskelijoita, joista kolmella ei ollut aiempaa kokemusta projektissa käytetyistä tietojärjestelmistä. Tämä alkuasetelma muodostui haasteelliseksi niin aikataululle kuin raporttikehitysmallillekin.

Diplomityön luvussa kaksi, määritellään termi ”*liiketoimintatiedon hallinta*”. Luvussa esitellään myös muut liiketoimintatiedon hallintaan liittyvät termit ja käsitteet. Luvussa kolme on esitetty liiketoimintatiedon hallinnan prosessi. Prosessi kuvaa kuinka organisaation prosesseista poimittu yksittäinen tieto päätyy useiden välivaiheiden kautta organisaation päätöksenteon tueksi. Luvussa neljä on esitetty 6pack-projektissa käytetyt ohjelmistot. Ohjelmistoista on kuvattu diplomityön kannalta oleelliset osuudet sekä ohjelmistojen rooli itse projektissa. Lukuun viisi on kerätty kolme erilaista ohjelmistokehitysmalleja, joiden parhaita puolia on pyritty hyödyntämään 6pack-projektissa. Lisäksi luvussa sivutaan ohjelmistokehitysprojektin resurssien valinnan teoriaa, kuinka valitaan tehokas ohjelmistokehitystiimi. Luvussa kuusi on kuvattu 6pack-projekti. Luvussa on esitetty projektin reunaehdot ja se, miten projekti toteutettiin. Luvussa seitsemän on arvioitu 6pack-projektin onnistumista. Projektin jälkeen haastateltiin kaikki projektiin osallistuneet. Luvussa on esitelty suoraan haastattelun tuloksia sekä tuloksista vedettyjä johtopäätöksiä ja arvioita. Luvussa kahdeksan on esitetty työn yhteenveto ja päätelmät, mitä projektista on seurannut.

2. LIKETOIMINTATIEDON HALLINTA

Business Intelligense -termi tarkoittaa suomeksi *liiketoimintatiedon hallintaa*. Mika Hannula määrittelee Business Intelligense -käsitteen seuraavasti (Hannula 2009):

Liiketoimintatiedon hallinnalla tarkoitetaan toimintaa, jossa kerätään liiketoimintaan ja liiketoimintaympäristöön liittyvää tietoa, tulkitaan ja analysoidaan sitä, arvioidaan tiedon merkitystä ja käytetään analysoitua tietoa päätöksenteon tukena.

Business Intelligensen tehtävänä on tukea organisaation päätöksentekoa. Pyrkimyksenä on saattaa oikea tieto oikeaan aikaan oikealla tavalla oikeaan paikkaan (Hannula 2009). Tavoitteena on organisaation entistä parempi suoritussyky. Oikea tieto tarkoittaa organisaatiossa liikkuvaa tietoa ja informaatiota, joka on jalostettu tietämykseksi ja ymmärrykseksi.

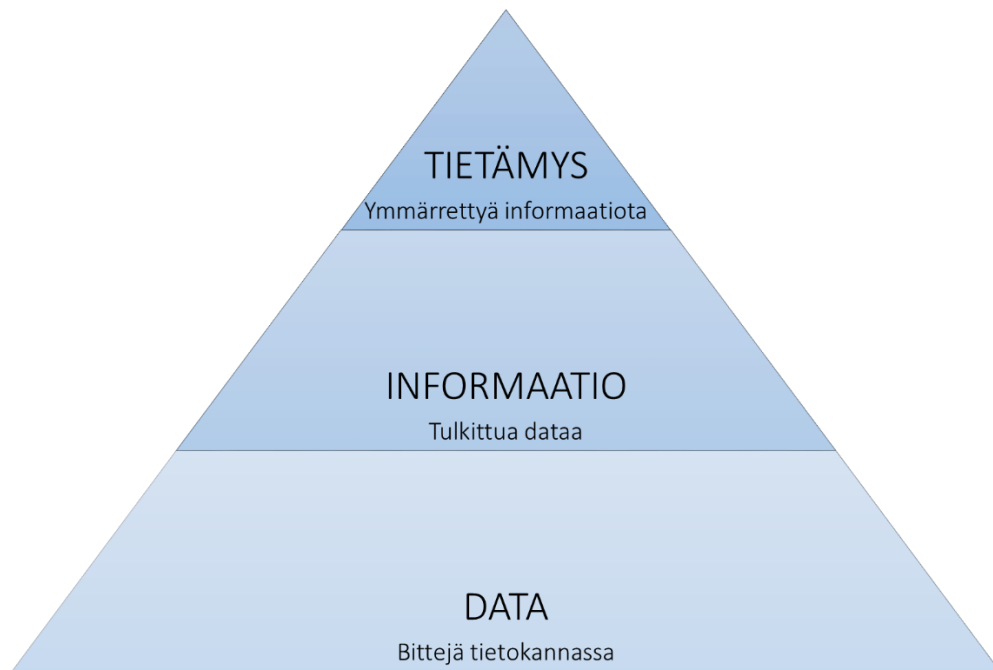
Oikea tieto ja tietoa tarvitsevat ihmiset on saatettava yhteen. Näin yrityksen tai organisaation henkilöstö, joka tuntee ja ymmärtää jalostetun tiedon ja informaation merkityksen sekä omaa ymmärryksen ja valtuudet tehdä tarvittavia päätöksiä, pystyy työskentelemään entistä tehokkaammin. Tiivistettynä Business Intelligensessä on kysymys ymmärryksen ja oivallusten tuottamisesta. Tämän luvun lähteenä on käytetty kandidaatintyötä (Aimo 2013).

2.1. Tieto

Organisaation hallussa oleva tieto on yksi tärkeimmistä kilpailutekijöistä ja siksi sitä pyritään vaalimaan. Tieto voidaan jakaa kolmeen perustasoon; **dataan**, **informaatioon** ja **tietämykseen**. Tiedon tasoja sekä tiedon jalostusarvon kasvua on havainnollistettu kuvassa 2.1.

Data tarkoittaa yksittäisiä tiedon jyvii, joita ei ole käsitelty tai jalostettu. Data on pääasiassa kvantitatiivista ja määrämuotoista, eikä se ole yksinään kovin hyödyllistä. Kun dataa tulkitaan, se muuttuu informaatioksi ja samalla hyödylliseksi (Virtanen 2011).

Informaatio on jalostettua dataa, jolla on jokin merkitys tarkastelijalleen. Informaatio ymmärretään usein datana, jolla on tietty tarkoitusperä. Dataa voidaan pitää siten tiedon esivaiheena. Data on välttämätöntä informaation ja **tietämyksen** saavuttamiseksi. (Lueg 2001).



Kuva 2.1 Tiedon tasot

Tietämys on informaatiota jalostuneempaa ja abstraktimpaa tietoa. Tietämys on johonkin asiaan liittyvä inhimillinen käsitys, joka saavutetaan kokemuksen, opiskelun ja ajattelun kautta (Virtanen 2011). Uuden tietämyksen tuottamiseksi tarvitaan informaation lisäksi aiempaa tietämystä. Tietämys liitetään usein myös asiantuntemukseen.

Kun tietoa tallennetaan, tallennetaan samalla myös tietoa määrittelevää tietoa, jota kutsutaan **metadataksi**. Tyypillisiä esimerkkejä metadatasta ovat esimerkiksi CD-levyn tiedot (levyn-nimi, esittäjä, säveltäjä, kappaleiden pituudet ja päivämäärät) ja digitaalikameroiden tallentamat tiedot valokuvien parametreista (kuvausajankohta, aukko, suljinnopeus jne.). Varsinainen tieto CD-levyllä on sen sisältämä musiikki tai digitaalikameran tapauksessa digitaalinen valokuva. Metadatan merkitys on kasvanut Internetin ja sosiaalisten medioiden suosion kasvun myötä. Metadatan avulla tietoa voidaan ryhmitellä, jolloin suuriin tietomassoihin kohdistuvia hakuja voidaan optimoida käyttämällä ryhmittelytietoon perustuvia algoritmeja. Metadata luo lisäksi edellytykset muun muassa versionhallinnalle.

2.2. Tiedon hankinta

Tieto on organisaation toiminnalle elinehto. Jos organisaatiossa ei tiedetä, miten sen prosessit tuottavat hyödykkeitä, on organisaation liiketoiminnan seuraaminen ja johtaminen vaikeaa. Siksi tiedon hankinta ja asianmukainen tallentaminen ovat keskeisiä organisaation toimintoja. Organisaatiot keräävät tietoa sekä organisaation sisältä että ulkopuolelta.

Organisaation sisäisiä tiedonlähteitä ovat erilaiset järjestelmät, jotka tukevat sen liiketoimintaprosesseja. Esimerkkeinä näistä ovat monenlaiset toiminnanohjausjärjestelmät. Ne huolehtivat tiedon tallentamisesta ja tiedon linkityksestä toisiinsa.

Kaikki organisaation prosessit eivät välttämättä kulje toiminnanohjausjärjestelmän kirjanpidon läpi. Näin voi käydä etenkin pienemmissä organisaatioissa, joissa ei ole lainkaan tietoa keräävää keskitettyä järjestelmää. Pienemmissä organisaatioissa on olennaista, että liiketoimintaprosesseja seurataan omatoimisesti ja niistä pidetään asianmukaista kirjanpitoa.

Ulkoista tietoa, johon katsotaan kuuluvan usein muun muassa kilpailijatieto, markkinatieto, teknologiatieto, tuotetieto, asiakastieto ja kumppanitieto, voidaan kerätä esimerkiksi Internetistä tai alan julkaisuista. Pienyrityksissä ulkoisen tiedon keruu ei aina ole kovin systemaattista. Mitä suuremmasta organisaatiosta on kyse, sitä todennäköisempää on, että ulkoisen tiedon kerääminen toteutetaan järjestelmällisesti, mahdollisesti erillisen osaston tai yksikön voimin. Ulkoisella tiedolla voidaan saavuttaa kilpailuetua muihin kilpailijoihin nähden, mikäli tulevaisuuden haasteisiin osataan varautua aiemmin ja tarvittaviin muutoksiin ryhdytään ennakoivasti.

2.3. Työkalut

Yksittäisen tiedon matka liiketoimintaprosesseista analyytikon päätöksenteon tueksi, on monivaiheinen prosessi. Prosessia suoraviivaistamaan on olemassa työkaluja. Toiminnanohjausjärjestelmät tallentavat automatisoidusti organisaation liiketoimintaprosesseista saadun tiedon tietovarastoihin. Jos automatisoituja järjestelmiä ei ole, tiedot kirjataan käsin. Tietovarastoista keskeinen tieto poimitaan myös automatisoidusti analyytikon raportille tai nykyisin yhä useammin liiketoimintatiedon hallintaa tukevaan järjestelmään. Liiketoimintatiedon hallintaa tukevissa järjestelmissä tietoa esitetään havainnollisesti ja siten, että oleellisen asian löytäminen olisi mahdollisimman vaivatonta.

2.4. Liiketoimintatiedon hallinnan tavoitteet

Liiketoimintatiedon hallinnalla organisaatio tavoittelee kilpailuetua. Organisaation toimintaa kehitettäessä on oleellista saada mahdollisimman aikaisessa vaiheessa laadukasta tietoa päätöksenteon tueksi. Liiketoimintatiedon hallinta auttaa organisaatiota ymmärtämään oman asemansa, asiakkaansa, kilpailijansa sekä sisäisen- ja ulkoisen toimintaympäristön.

Keskeinen menestyksen mittari on toiminnan oikea-aikaisuus. Organisaation on osattava toimia oikeaan aikaan ja oikealla tavalla, ennen kilpailijoita. Organisaation on osattava kyseenalaistaa olemassa olevaa tietoa ja kyettävä valitsemaan ketterämpiä ja

tehokkaampia tapoja toteuttaa asioita, jotta uusille oivalluksille luodaan riittävät edellytykset. Organisaation päätöksenteosta saadaan laadukkaampaa, kun tietoa osataan käsitellä uusien menetelmin ja hyödyntää saavutettuja tuloksia oikealla tavalla. Näin organisaation suorituskyky paranee, kilpailukyky kasvaa ja kustannukset alenevat. Liiketoimintatiedon hallinnan tavoitteita ovat tulevaisuuden liiketoiminnan hallitsemisen oppiminen, reagointiajan parantaminen, aineettoman pääoman suojeleminen, yllätysten välttäminen ja mahdollisuuksien sekä uhkien tunnistaminen. (Moss & Atre 2003)

Teknisestä näkökulmasta liiketoimintatiedon hallinta on kokonaisuus, jossa yhdistyvät liiketoimintatiedon dynaaminen analysointi ja tiedon louhinta siten, että näitä sovelletaan yhdessä tietovarastoihin säilötyn tiedon käsittelyssä.

3. LIKETOIMINTATIEDON HALLINTA PROSESSINA

Tiedon prosessointi koostuu erilaisista tekniikoista ja vaiheista. Käytetyt menetelmät ja tekniikat vaihtelevat organisaation tarpeiden mukaan. Pienissä organisaatioissa tiedon prosessointi saattaa puuttua kokonaan. Tämän luvun lähteenä on käytetty kandidaatintyötä (Aimo 2013).

Tiedon prosessoinnissa on kyse tiedon muuttamiseen käyttäjälle sopivaan muotoon ja tiedon saattamisesta käyttäjän ulottuville. Prosessointi käynnistyy yksittäisen tiedon haulla organisaation sisäisistä tai ulkoisista tietolähteistä. Haettua tietoa muokataan ETL-prosessissa, jonka seurauksena muodostuu jalostettu tietovarasto. Tietovarasto voidaan muodostaa paikallisvarastoista, jotka sisältävät vain osan tietovarastosta. Tietovarastosta tai paikallisvarastoista tietoa noudetaan erilaisilla työkaluilla ja esitetään raporteina tai kaavioina. Kuva 3.1 havainnollistaa edellä kuvattua prosessia.

3.1. Tietokannat

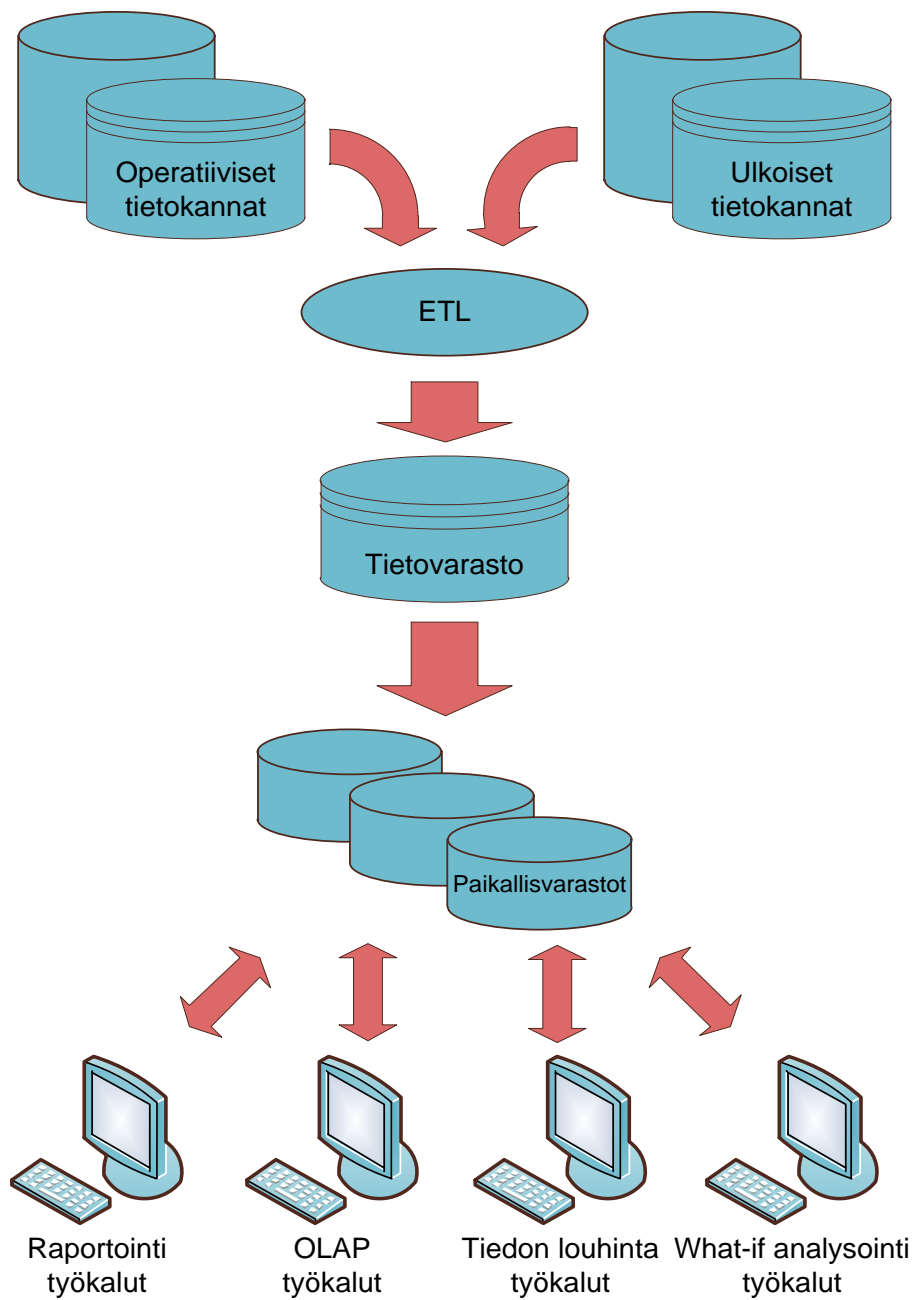
Tietokanta on paikka, johon tieto tallennetaan fyysisesti. Se on tiedon säilytyspaikka, jota useat käyttäjät voivat käyttää samanaikaisesti. Tietokanta mahdollistaa tiedon varman säilymisen. Tietokannassa on käytössä yhtenäinen tallennuskerros, jota erilaiset järjestelmät käyttävät. Tietokanta on toteutettu siten, että siellä on mahdollisimman vähän kahdennettua tietoa. Tietokantaan tallennetaan organisaation operatiivisen järjestelmän tietojen lisäksi myös metadataa. (Begg & Connolly 2005).

Käyttäjät ovat vuorovaikutuksessa tietokantojen kanssa sovellusohjelmien välityksellä. Niillä muokataan, ylläpidetään ja luodaan uutta tietoa tietokantaan.

3.2. Tietovarastot

Tietovarastot muodostuvat tietokannoista. Tietovarasto on keskitetty tiedon tallennuspaikka, johon liiketoimintaprosesseissa liikkuvaa tietoa säännöllisesti kerätään. Tietovarastossa tieto organisoidaan uudelleen, validoidaan, järjestetään ja sitä täydennetään muiden tietolähteiden tiedoilla (Jacot et al. 2009). Kun tietovarastoon kerätään tietoa useista eri tietokannoista, saadaan tietovarastosta organisaation henkilöstön pääasiallinen tietolähde. Analyttikot keräävät tietovarastosta tarvittavia tietoja, joilla he tukevat päivittäistä työtään.

Tietovarastojen käyttö tuo organisaatiolle liiketoiminnallisia etuja. Kun tieto on keskitetty yhteen paikkaan, sitä ei tarvitse etsiä eri tietolähteistä. Tietovarastossa tieto on lisäksi yhtenäistä ja ajantasaista, joten tietovarastojen käyttäjät voivat luottaa saavansa ajantasaisen tiedon tietovarastosta.



3

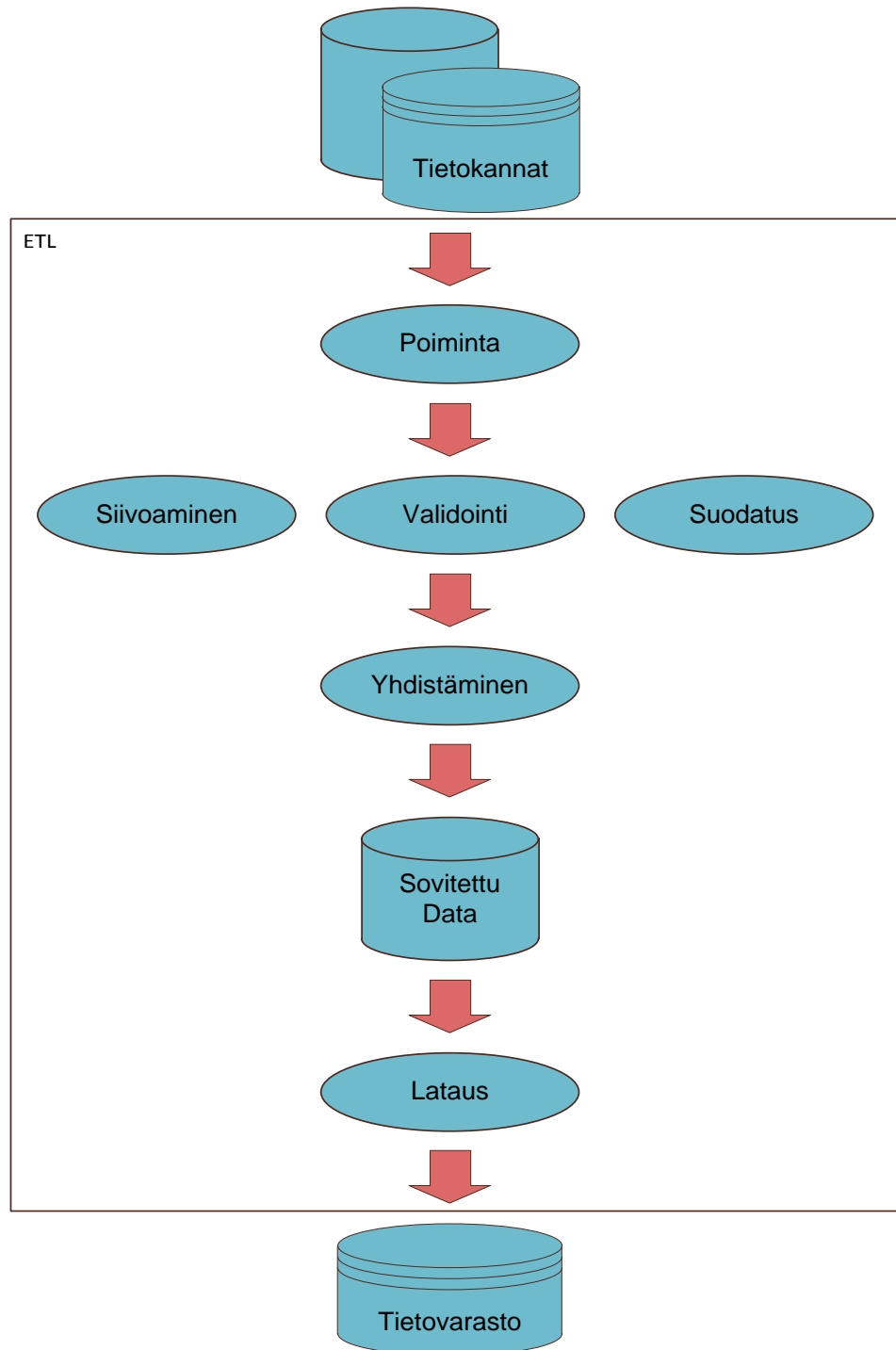
Kuva 3.1 tiedon kulku organisaatiossa

3.2.1. Paikallisvarastot

Paikallisvarasto tarkoittaa yleensä tietovarastosta johdettua pienemmälle, esimerkiksi yhdelle osastolle suunnattua tietokantaa (Golfarelli & Rizzi 2009). Paikallisvarastojen käyttö ei ole pakollista. Mikäli paikallisvarastoja käytetään, voidaan yhteen paikallisvarastoon tallentaa yhden liiketoimintayksikön tai organisaation osaston tarvitsemat tiedot. Paikallisvarastoilla voidaan tehostaa tietovaraston käyttöä, kun organisaation yksikkö suorittaa kyselyjä vain tiettyyn paikallisvarastoon.

3.2.2. ETL

ETL on prosessi, jossa tietoa poimitaan operatiivisista tietokannoista, muokataan raportointia varten ja ladataan tietovarastoon. ETL tulee englannin kielen sanoista Extract, Transform, Load. ETL –prosessi on esitetty kuvassa 3.2.



Kuva 3.2 ETL-prosessi

Prosessin alussa operatiivisista tietokannoista poimitaan tarvittava osa tietoa. Prosessin edetessä poimitulle tiedolle suoritetaan tarpeen mukaan erilaisia muunnos-toimenpiteitä.

Tietoa voidaan siivota, rajata, validoida ja yhdistää. Erilaisten toimenpiteiden selitykset menevät kirjallisuudessa usein sekaisin. Seuraavissa kappaleissa esitelty toimenpiteet noudattelevat kirjassa *Data Warehouse Design: Modern Principles and Methodologies* (Golfarelli & Rizzi 2009) esitettyä mallia. Kun tarvittavat tiedon siivous- ja yhtenäistämistoimet on tehty, ladataan tieto tietovarastoon.

Muunnokset. Muunnosvaihe on ETL -prosessin tärkeimpiä. Sen tarkoituksena on parantaa tiedon laatua, joka on usein melko heikotasoista lähdejärjestelmistä riippuen. Muunnokset toteutetaan puhdistus- ja homogenisointityökaluilla, jotka hyödyntävät erilaisia kirjastoja. Kirjastoissa on eritelty esimerkiksi eri sanojen synonyymejä sekä tyypillisimpiä kirjoitusvirheitä. Muunnosvaiheessa datasta pyritään poistamaan duplikaatteja, mahdottomia ja epäjohdonmukaisia arvoja, jotka ovat loogisesti kytköksissä muuhun tietoon.

Esimerkki mahdottomasta arvosta on päivämäärä 30.2.2009. Joitakin puuttuvia tietoja saatetaan lisätä, jos voidaan olla varmaoja lisättävien tietojen oikeellisuudesta. Jos tietoja on kytketty toisiinsa väärin, esim. sosiaaliturvatunnus on asetettu puhelinnumeroita sisältävään sarakkeeseen, voidaan edellä mainittuja epäjohdonmukaisuuksia korjata. Muunnosvaiheessa poistetaan myös kirjoitusvirheitä, ja yhdistetään tarvittaessa samoja tietoja. Esimerkiksi ihmisen ikä saattaa olla ilmoitettu numeroilla tai kirjaimilla. Tiedon yhtenäistäminen parantaa tiedon luettavuutta.

Lataus. Latausvaiheessa data tallennetaan tietovarastoon tai paikallisvarastoon. Lataus voidaan tehdä kahdella tavalla. Vaihtoehto yksi on poistaa ensin tietovarastosta ole olemassa oleva data ja ladata uusi data. Toisessa vaihtoehdossa päivitetään jo olemassa olevaa dataa uudella datalla. Yleensä pyritään siihen, että jo kertaalleen ladattua dataa ei koskaan tarvitsisi muuttaa tai ladata uudestaan. On kuitenkin väistämätöntä, että jossain vaiheessa data joudutaan lataamaan alusta uudestaan, joten siihenkin on hyvä varautua (Kimball & Caserta 2004).

3.3. Raportointi

Raportit ovat organisaation johdon operatiivisen toiminnan perustyökalu liiketoiminnan suunnitteluun ja arviointiin. Tietovarastointi on tuonut raportointiin oman lisänsä. Kun tietovarastoja käytetään lähteenä, raportin dataan voidaan luottaa paremmin. Lisäksi tietovarastot parantavat raporttien suorittamisen tehokkuutta, sillä tietovarastossa suurin osa laskettavista asioista on laskettu jo valmiiksi. Näin itse raporttien suorittaminen on nopeaa.

Raportointi voi tarkoittaa montaa asiaa. Tässä asiayhteydessä sillä tarkoitetaan lähes staattista informaation lähestymistapaa. Raportoinnilla tarkoitetaan raporttien suunnittelua ja niiden suorittamista. Raportit ovat yleensä tulosteita, esimerkiksi kuukausittaisia tulosraportteja. Raporteille on tyypillistä, että niiden ulkoasu on

tarkkaan määrätty. Raportin ulkoasua saattavat rajata lisäksi erilaiset lakipykälät ja säädökset, esimerkkinä pörssiyritysten tilinpäätökset. Raportit ovat kuvainnollisesti jäädytettyjä näkymiä organisaation järjestelmissä liikkuvaan informaatioon. Käyttäjillä on tavallisesti mahdollisuus tuottaa raportti valitsemillaan parametreilla, jolloin voidaan ohjata raportin taustalle rakennettuja tietolähteisiin kohdistuvia kyselyitä. Tämä lisää raportin monikäyttöisyyttä.

Raportti koostuu tiedon kyselystä ja ulkoasun asetelmasta. Kyselyllä tarkoitetaan niitä summauksia, ehtoja ja rajoituksia, joilla muodostetaan raportin tietosisältö. Ulkoasu kuvaa, miten eri asiakokonaisuudet raportilla esitetään.

Raportti voidaan muodostaa joko käsin käyttäjän toimesta tai esimerkiksi ajastetusti palvelimen suorittamana. Palvelin voidaan määrittää lähettämään valmiit raportit suoraan ennalta määrättyjen henkilöiden sähköpostiin tai verkkolevylle.

3.4. What-if

What-if -analyysi eli herkkyysanalyysi on simulointia, jossa tutkitaan lähtöoletusten muuttamisen vaikutuksia systeemiin (Golfarelli & Rizzi 2009). Herkkyysanalyysillä voidaan vertailla, millä muutosarvoilla saadaan haluttu lopputulos, kuinka paljon tiettyä arvoa tai arvojoukkoa tarvitsee muuttaa ja mihin suuntaan, jotta lopputulos on tavoitteiden mukainen.

Herkkyysanalyysillä pyritään löytämään vastauksia muun muassa seuraavan tyyppisiin kysymyksiin (Golfarelli & Rizzi 2009):

1. Jos jauhojen pakkaushintaa alennettaisiin viidellä prosenttiyksiköllä kahden kuukauden ajaksi, kuinka paljon hiivan myynti muuttuisi?
2. Mitä tapahtuisi, jos organisaation budjetti jaettaisiin uudella tavalla eri osastojen välillä?
3. Miten henkilöstökustannukset muuttuisivat seuraavan viiden vuoden aikana, jos uusi rekrytointiohjelma otettaisiin käyttöön?

On olemassa useita erilaisia herkkyysanalyysityyppejä. *Osittaiserhkyysanalyysissa* muutetaan yhden parametrin arvoa ja approksimoidaan sen vaikutuksia muuhun systeemiin. Malli ei anna kattavaa kuvaa koko systeemin herkkyydestä, mutta kuvaa muutettavan parametrin kriittisyyttä. *Parhaan ja huonoimman tapauksen analyysissa* valitaan lähtöoletukset parametriarvojen ääripäistä. Ero osittaiserhkyysanalyysiin on siinä, että menetelmässä muutetaan samalla useita lähtöoletuksia. Mallilla voidaan testata, muuttuvatko tarkasteltavan systeemin tulokset päinvastaisiksi valituissa ääritapauksissa.

4. KÄYTETYT OHJELMISTOT

Tässä työssä tutkitussa kehitysprojektissa, käytettiin pääasiallisesti kahta ohjelmistoa. QlikView-ohjelmistoa käytettiin raporttien tuottamiseen ja Dynamics Ax -ohjelmiston tietokantaa käytettiin raporttien tiedon lähteenä. Dynamics Ax -ohjelmistoa käytettiin lisäksi raporttien laadun varmistuksessa ja tiedon oikeellisuuden tarkistuksissa.

4.1. QlikView-ohjelmisto

QlikView-ohjelmisto on Qlikin valmistama ja myymä liiketoimintatiedon hallintajärjestelmä. Järjestelmä tarjoaa perinteisten staattisten raporttien lisäksi dynaamisen käyttöliittymän tietojen analysointiin.

4.1.1. Qlik

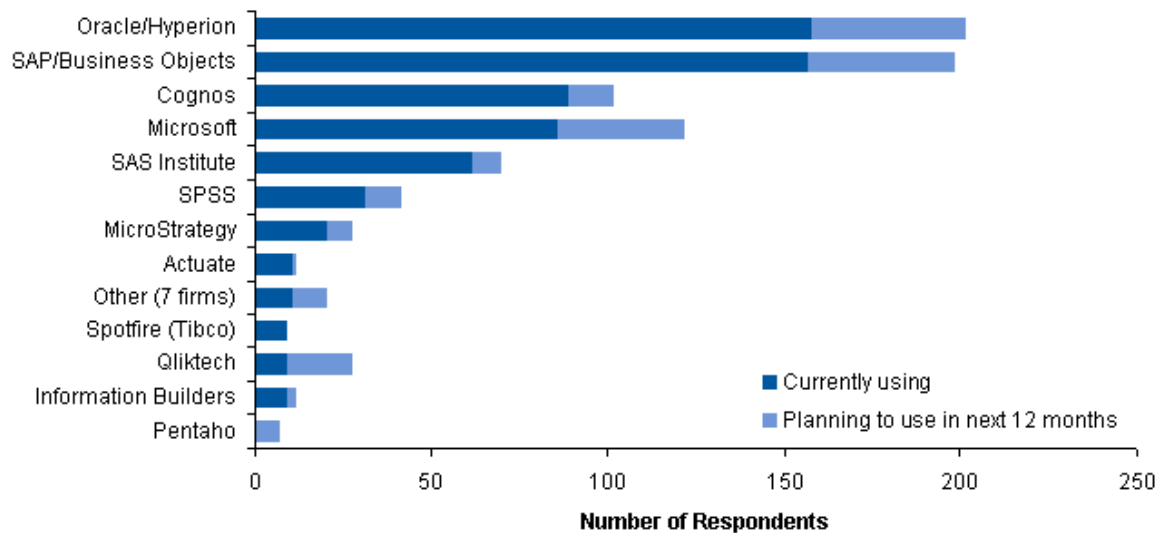
Qlik, entinen QlikTech International, on 1993 Ruotsissa perustettu konsultointiyritys. Qlik on keskittynyt raportointityökalujen kehittämiseen. Yrityksellä oli vuonna 2009 11400 asiakasta. Qlikin pääkonttori sijaitsee Yhdysvalloissa Pennsylvaniassa. Qlikin ensimmäisellä tuotteella QlikViewillä on yli puoli miljoonaa käyttäjää. Vuonna 2014 Qlik julkaisi toisen raportointijärjestelmän, Qlik Sense. Qlik Sense on mobiilikäyttöön suunnattu raportointijärjestelmä, joka hyödyntää samaa moottoria kuin QlikViewkin.

Vuonna 2008 Qlik oli nopeimmin kasvava Business Intelligence -yritys yli 50% kasvuvauhdillaan (Qlik, 2009). Qlikin voimakkaan kasvun taustalla yksi merkittävimmistä syistä on sen tuotteiden, QlikViewin ja Qlik Sense alhaiset kehitys- ja käyttöönottokustannukset.

Qlik markkinoi QlikViewiin liittyen konseptia, josta käytetään nimitystä SIB (Seeing Is Believing). Siinä asiakasyritykselle räätälöidään muutamassa päivässä malliraportti asiakkaan kriittisimmistä ongelmakohdista. Asiakkaalla on tämän jälkeen kaksi viikkoa aikaa tutustua malliraporttiin sekä QlikView-tuotteeseen. Konseptia varten Qlik lainaa asiakkaalle lisenssit, eikä peri niistä erillistä maksua. Konseptilla pyritään varmistamaan QlikViewin soveltuvuus asiakasyrityksen tarpeisiin. Nykyisessä taloustilanteessa SIB-konseptilla on paljon kysyntää, sillä analyysiä ja ennustetta tulevaisuuden näkymistä pitäisi saada tuotettua ilman suuria IT-kustannuksia. (Richardson et al. 2009)

Vahvan kasvun lisäksi tutkimukset osoittavat Qlikin nauttivan korkeaa asiakastyytyväisyyttä. Business Intelligence -tuotteiden vertailuun keskittynyt Gartner on teettänyt kyselyn, jossa asiakkailta kysyttiin heidän näkemystään Qlikistä ja Qlikin onnistumista asiakkaan Business Intelligence -projektissa. Kysely osoitti, että asiakastyytyväisyys oli parempi kuin millään muulla Business Intelligence -palveluntarjoajalla. (Richardson et al. 2009)

Vaikka Qlikin liikevaihto kasvoi vuonna 2008 voimakkaasti, on yrityksellä vielä matkaa kurottavanaan, mikäli se mieli kilpailla alan suurimmista asiakkaista. Vuoden 2009 Business Intelligence kilpailutilanteesta Qlik löytyi sijalta 10. Tämä perustuu Gartnerin (Richardson 2009) teettämään kyselyyn. Kyselyyn osallistui yli 200 yritystä ja siinä kysyttiin esimerkiksi, mitä Business Intelligence -työkalua yritys käyttää tai aikoo käyttää seuraavan 12 kuukauden aikana. Qlikin edellä olivat mm. Oracle, SAP, Cognos ja Microsoft. Alan yritysten keskinäinen kilpailuasetelma näkyy kuvassa 4.1.



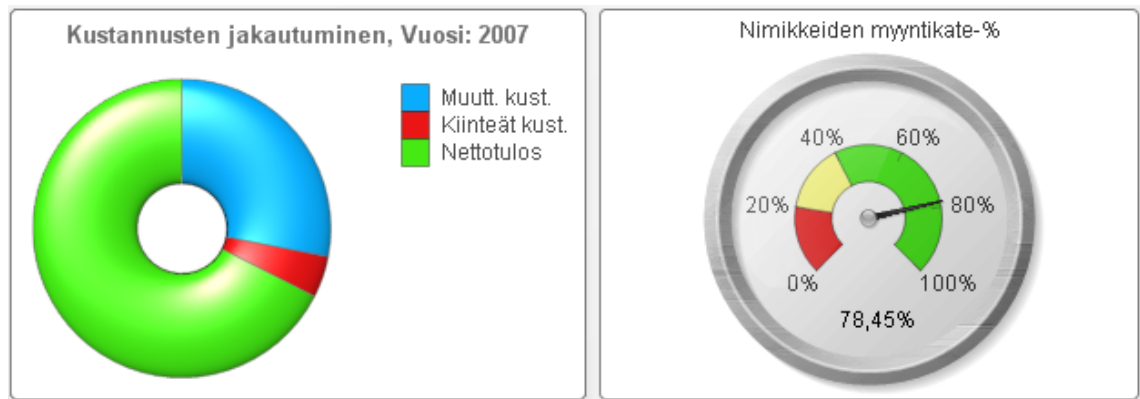
Kuva 4.1 Adoption of BI Platform Vendors (Richardson 2009)

4.1.2. QlikView

QlikView on Qlikin kehittämä muistipohjainen raportointityökalu. Yrityksen yksi ensimmäisistä työtehtävistä oli kehittää moniulotteisen tiedon analysointityökalu. Ensimmäinen PC-versio QlikViewstä (versio 3.15) julkaistiin vuonna 1997.

QlikViewssä kaikki käyttäjän tarvitsema tieto pakataan ja ladataan joko käyttäjän oman koneen RAM-muistiin tai palvelimen RAM-muistiin, riippuen raportin avausmenetelmästä. Tiedon lataus toteutetaan suorittamalla käyttäjän kirjoittama skripti, joka koostuu SQL-kyselykielen mukaisista hakulauseista sekä QlikViewn sisäisesti määritellyistä funktioista. Skriptin suorittamisen pohjalta QlikView muodostaa raportin datamallin. Datamalli kuvaa tietorakennetta, joka on tallennettu joko käyttäjän koneelle tai palvelimelle. Skripti ja datamalli on esitelty tarkemmin kohdassa 4.1.3.

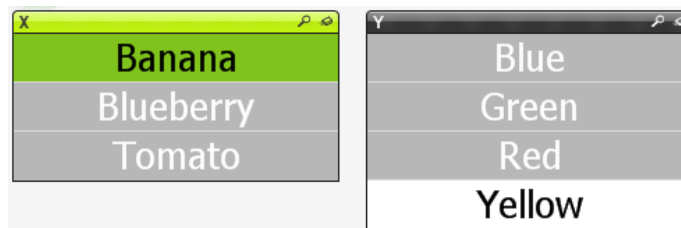
Käyttäjä voi analysoida datamallin tietoja vapaasti haluamallaan objekteilla, joita QlikViewn versiosta 11.2 löytyy 14 erilaista. Objekti tarkoittaa QlikView-raportissa olevaa elementtiä. Objekti on kokonaisuus, joka piirretään raportille annettujen rajojen sisäpuolelle. Tiedon analysointi tapahtuu objekteja tulkitsemalla. Kuvassa 4.2 on esitetty kaksi QlikView-raportille tyypillistä objektia, Pie Chart ja Gauge Chart.



Kuva 4.2 QlikView Pie Chart ja Gauge Chart.

QlikView-objekteissa on pyritty panostamaan niiden visuaalisuuteen ja interaktiivisuuteen. Esimerkiksi kuvaajista ja suodattimista voi tehdä suodatuksia.

Suodatuksella tarkoitetaan tiedon rajaamista. Sen ymmärtäminen selviää helpoiten esimerkin avulla. Suodattamalla käyttäjä rajaa kuvaajissa ja muissa objekteissa näytettäviä tietoja. Kuvassa 4.3 on esillä QlikViewille tyypillinen objekti; Listbox. Listbox on laatikko, jossa on listattuna yhden kentän sisältämiä arvoja, tässä tapauksessa kenttien X ja Y arvoja.



Kuva 4.3 QlikView listbox suodattimet (X ja Y).

Kenttä voidaan ymmärtää Excel-taulukon sarakkeena. Kentän arvot löytyvät kuvan 4.4 riveiltä 1-4. Kuvassa 4.3 on X-kentän Listboxista suodatettu valinta "Banana". Vihreä väri kuvaa oletuksena QlikViewissä suodatusta. X ja Y -kentillä on suora yhteys QlikViewn datamallissa. X-kentän suodatus vaikuttaa näin myös Y-kenttään rajaten pois sieltä sellaiset arvot, joilla ei ole assosiatiivisuutta X-kentässä tehdyn suodatuksen kanssa, esimerkissä arvot "Blue", "Green" ja "Red". Arvo "Yellow" sen sijaan on valkoisella pohjalla, koska se liittyy tehtyihin valintoihin.

Book1		
	X	Y
1	Banana	Yellow
2	Blueberry	Blue
3	Tomato	Green
4	Tomato	Red

Kuva 4.4 Datamalli Excel-taulukkona.

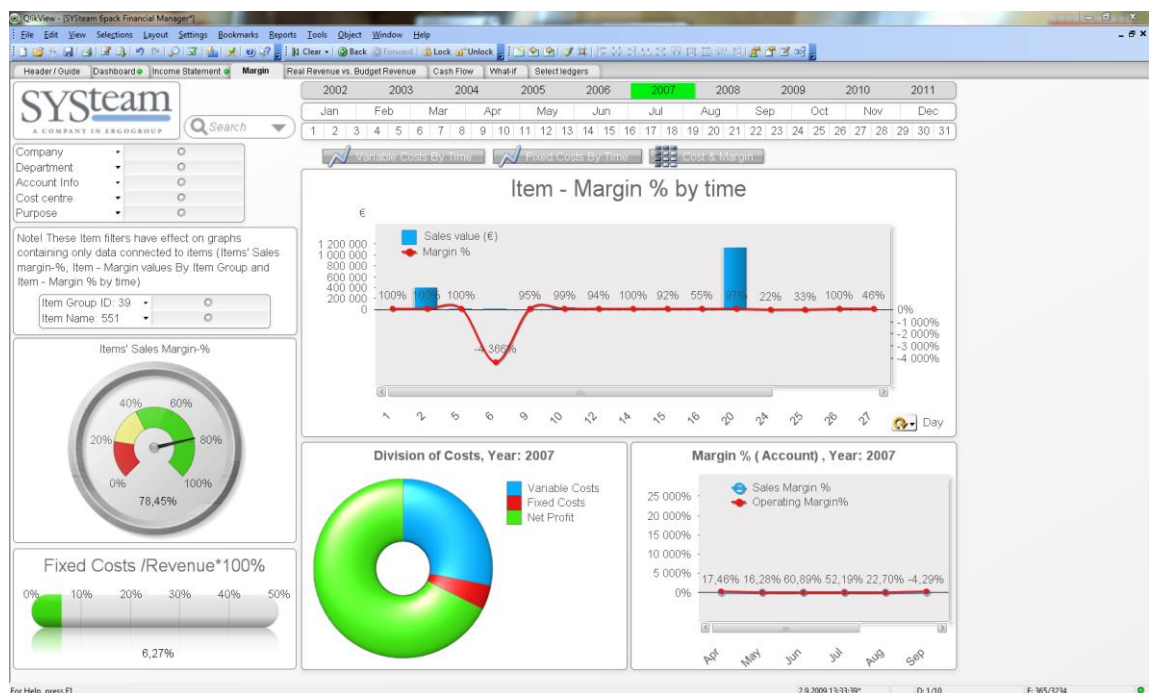
Suodatuksia voi tehdä yhden tai useita. Usein toistuvia suodatuksia tai suodatussarjoja voi tallentaa kirjanmerkeiksi. Kirjanmerkki on tallennettu joukko suodatuksia. Kun kirjanmerkki otetaan käyttöön, siihen tallennetut suodatukset toteutetaan. Kuvassa 4.5 päivät 12, 15 ja 22 on valittu suodattimiksi.

Suodatuksia voi halutessaan myös lukita. Kuvassa 4.5 on kuukaudet maaliskuu- ja huhtikuu lukittu, jolloin suodatukset pysyvät päällä vaikka muut suodatukset poistettaisiin tyhjentämällä kaikki suodatukset. Lukitut suodatukset voi poistaa ainoastaan avaamalla lukon ja tämän jälkeen poistamalla lukosta avatut suodatukset. Sininen väri kuvaa QlikView:ssä lukittua suodatusta.

2002			2003			2004			2005			2006			2007			2008			2009			2010			2011								
Tammi			Helmi			Maalis			Huhti			Touko			Kesä			Heinä			Elo			Syys			Loka			Marras			Joulu		
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31					

Kuva 4.5 päivämäärään liittyviä suodattimia.

QlikView:ssä raporttien visuaalisuus on räätälöitävissä. Lähes kaikkea, mitä raportti pitää sisällään, voi värittää ja asetella haluamallaan tavalla. Kuvassa 4.6 on esimerkki QlikView-raportista. Raportista voidaan havaita selkeät kirkkaat värit, eri värien osuudet ja poikkeamat keskiarvosta. Poikkeamilla ja niiden havainnoinnilla on tiedon analysoinnissa yleensä tärkeä merkitys. Niihin halutaan puuttua mahdollisimman aikaisessa vaiheessa. QlikView sisältää tätä silmälläpitäen lukuisia vaihtoehtoja poikkeamien esittämiseen ja eriyttämiseen muusta tiedosta.



Kuva 4.6 QlikView-raportti – Talousjohtaja/6pack.

Käyttäjän näkökulmasta QlikView on pyritty tekemään mahdollisimman helppokäyttöiseksi, jotta uuden järjestelmän oppimisesta ei koituisi kynnystä uutta työkalua valittaessa. Samasta syystä järjestelmän toiminnallisuus on tuotu lähelle Microsoftin toimisto-ohjelmien toiminnallisuutta. Microsoftin toimisto-ohjelmia käyttänyt henkilö omaksuu uuden QlikView-ohjelmiston toiminnallisuudet nopeasti (Qlik 2009).

4.1.3. QlikView-raportit

QlikView-raportit koostuvat yhdestä tai useammasta KPI-välilehdestä, dashboard-välilehdestä sekä tarkempaan rivi-analysointiin keskittyvistä analyysi-välilehdistä. KPI (Key Performance Indicator) tarkoittaa mittaria, joka ilmaisee suorituskyyä jonkin tavoitearvon suhteen; esimerkiksi *myynti*, *kate* tai *varastonkiertonopeus*.

Raportin jaottelu eri välilehdille parantaa havainnollisuuden ja selkeyttämisen lisäksi raportin ylläpidettävyyttä ja uudelleenkäytettävyyttä. Yhdelle välilehdelle asetellut yhtä KPI:tä koskevat objektit ja suodatimet voidaan kopioida toiseen raporttiin, jolloin asiakas voi valita rooliaan koskevat KPI:t, eikä räätälöintiin kulu ylimääräistä aikaa.

Dashboard-välilehti on kokoelma organisaation keskeisimmistä KPI-mittareista. Sen tarkoitus on antaa yhdellä silmäyksellä kattava kuva siitä, mikä on organisaation hetkellinen tilanne. Dashboard-välilehdeä ei usein löydy yhtään suodatinta.

Analyysi-välilehdillä on taulukoin ja kuvaajin mahdollistettu tietojen tarkempi tutkiminen. Analyysi-välilehdiltä voi käyttäjä etsiä vastauksia Dashboardilla olevista KPI-mittareista ilmenneisiin kysymyksiin. Analyysi-välilehtien taulukot helpottavat myös tiedon tarkistamista ja virheellisten kirjausten löytämistä muusta tiedosta.

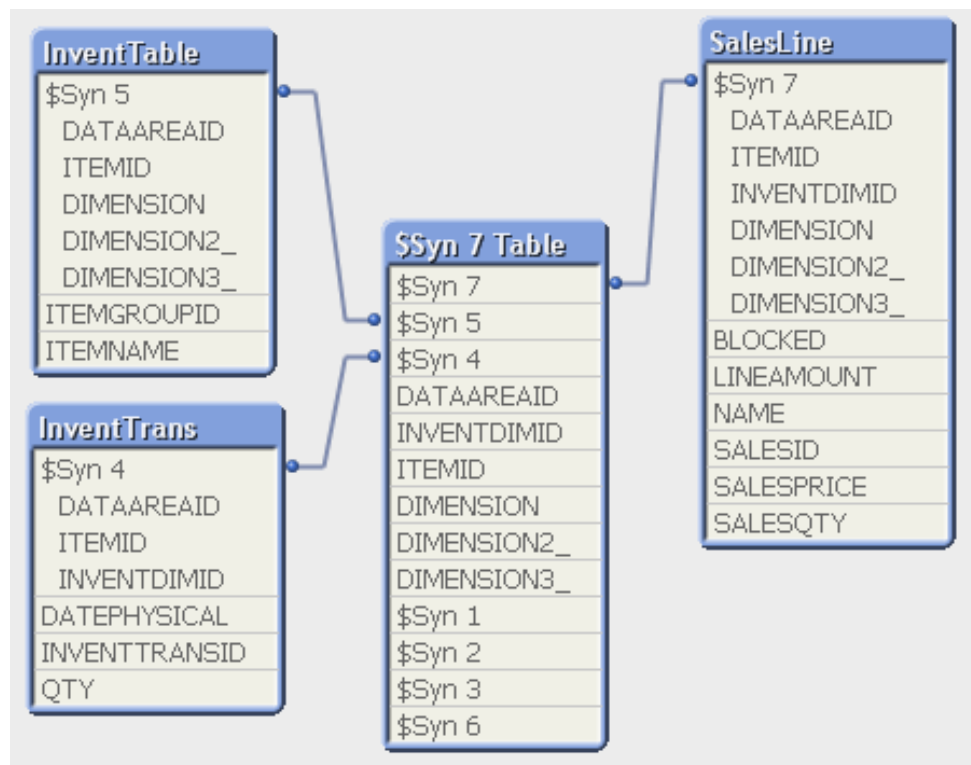
Kaikki QlikView-raportilla esitettävä tieto on tallennettu Qlikin patentoimalla tavalla raportin sisälle. Tämän vuoksi raporttia käytettäessä ei tarvita yhteyttä tietovarastoon tai tietokantaan. Raportin sisältämä tieto on yhtä tuoretta, kuin se oli raportin viimeisellä päivityskerralla. Tiedon haussa QlikView hyödyntää käyttäjän kirjoittamaa skriptiä, jonka QlikView suorittaa joko ajastetusti tai käyttäjän antamalla komennolla. Skriptin pohjalta QlikView muodostaa datamallin. Tämän perusteella QlikView tallentaa ladatut tiedot raportin sisään.

QlikView-datamalli koostuu tietoa sisältävistä tauluista ja taulujen välisistä relaatioista. Relaatio tarkoittaa yhteyttä kahden tai useamman taulun välillä. Muodostamalla relaatioita eri taulujen välille luodaan samalla rakenne, jossa valitsemalla taulusta jokin erityinen kiinnostuksen kohde voidaan relaatioita pitkin kulkea muihin tauluihin ja tarkastella kiinnostuksen kohteeseen liittyviä muiden taulujen tietoja. Relaatioiden muodostuminen on QlikView-datamallissa automaattista. Relaatio kahden taulun välille syntyy, kun molemmissa tauluissa on samanniminen kenttä. Käyttäjä voi kuitenkin

estää relaatioiden muodostumisen taulujen välille nimeämällä taulujen kentät eri tavalla. Taulujen välisen relaation muodostavaa kenttää kutsutaan avainkentäksi.

QlikView-datamalli sallii kahden taulun välille vain yhden relaation, joten silmukkarakenteet eivät ole sallittuja. Mikäli silmukkarakenteita kuitenkin syntyy, QlikView pyrkii ratkaisemaan ne automaattisesti muodostamalla taulujen välille erillisen, synteettisillä avaimilla varustetun taulun. Tämä taulu toimii linkkinä varsinaisten taulujen välillä ja poistaa samalla silmukkarakenteen. Synteettinen avain on avain, joka sisältää useamman kuin yhden kentän.

Kuvassa 4.7 QlikView on muodostanut synteettisen taulun *\$Syn 7 Table* taulujen *InventTrans InventTable* ja *SalesLine* väliin. Taulujen väliset relaatiot on muodostettu *\$Syn 4*, *\$Syn 5* ja *\$Syn 7* kentillä siten, että kukin taulu on yhteydessä toiseen vain yhdellä avainkentällä. Synteettiset avainkentät sisältävät kuvassa ilmenevien kenttien sisällön.



Kuva 4.7 Datamalli, jossa synteettinen taulu ja synteettisiä avaimia.

QlikView-skriptissä käyttäjä määrittää, mitä tietoa raporttiin tuodaan, kuinka pitkältä ajalta, mihin muuhun tietoon nyt tuotava tieto liitetään ja missä muodossa tieto tallennetaan. Lisäksi skriptissä määritetään paikat, mistä tietoa haetaan ja mitä yhteyksiä eri tietolähteillä on. QlikView-skripti on raportin tapaan jaoteltu eri välilehdille ylläpidettävyyden ja luettavuuden parantamiseksi.

Kuvassa 4.8 on osa QlikView-skriptiä. Se sisältää erään taulun tuontilauseen. Skriptissä rivillä 53 mainitaan, että tiedot haetaan *AxTesti*-tietokannasta sen

CUSTINVOICEJOUR-taulusta. Ylimääräisten relaatioiden syntymisen estämiseksi, on tyypillisimmät kentät nimetty *Sales*-tauluun viittaavalla etuliitteellä, kuten rivillä 37 *SFI_SalesOwnID*.

```

32 // Sales Invoices
33 Sales:
34 LOAD lower(DATAAREAID) as SFI_SalesDataAreaID,
35     DUEDATE as SFI_DATE,
36     INVOICEAMOUNTMST as SFI_SalesAmount,
37     INVOICEID as SFI_SalesOwnID,
38     SALESID as SFI_SalesReferenceID,
39     // Flag for 'normal' order invoices
40     'CUSTINVOICE' as SFI_SalesFlag,
41     DIMENSION,
42     "DIMENSION2_",
43     "DIMENSION3_";
44 SQL SELECT DATAAREAID,
45     DUEDATE,
46     //INVOICEAMOUNT,
47     INVOICEAMOUNTMST,
48     INVOICEID,
49     SALESID,
50     DIMENSION,
51     "DIMENSION2_",
52     "DIMENSION3_"
53 FROM AxTesti.dbo.CUSTINVOICEJOUR;

```

Kuva 4.8 CUSTINVOICEJOUR-taulun tuontilause QlikView-skriptissä.

4.2. Tietojärjestelmä

Toinen integraatioprosessissa käytetyistä järjestelmistä on Microsoftin Dynamics -tuoteperheeseen kuuluva, alkujaan Tanskalaisen Damgaard ohjelmistotalon toteuttama Axapta-tietojärjestelmä, nykyisin nimeltään Dynamics Ax.

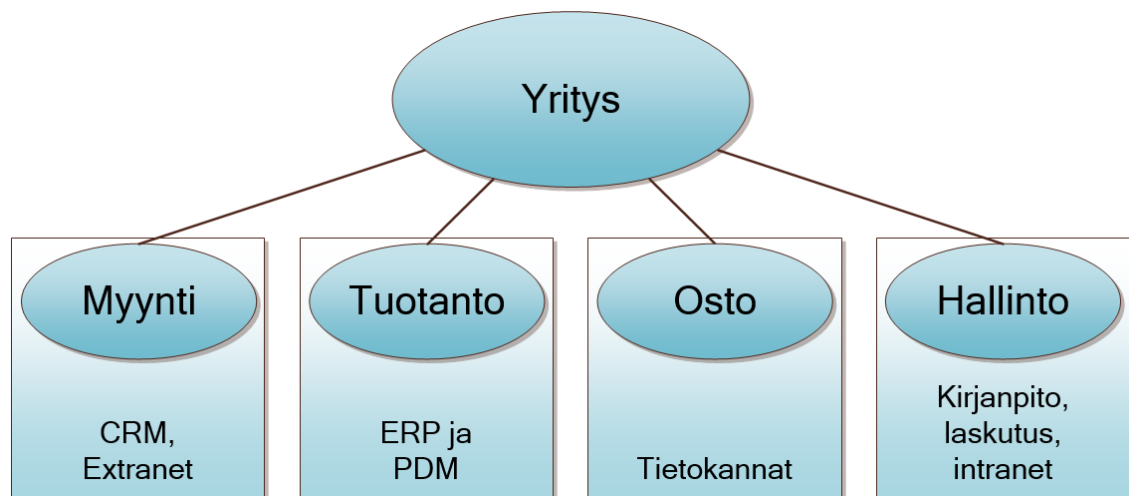
Sami Kettunen on kirjassaan Tietojärjestelmän ostaminen (Kettunen 2002) määritellyt tietojärjestelmän viitaten ATK-sanakirjaan vuodelta 1999:

Tietojärjestelmällä tarkoitetaan ihmisistä, tietojenkäsittelylaitteistoista, tiedonsiirtolaitteistoista ja ohjelmista koostuvaa järjestelmää, jonka tarkoitus on tietoja käsittelemällä tehostaa tai helpottaa jotakin toimintaa tai tehdä toiminta mahdolliseksi.

Kettunen on jaotellut tietojärjestelmiä organisaation eri osastoja varten. Organisaation koon ja toimialan mukaisesti siitä on tavallisesti löydettävissä ainakin hallinnon osasto. Osasto voi olla harhaanjohtava, sillä pienimmissä organisaatioissa hallinnosta vastaa usein yksi henkilö, organisaation toimitusjohtaja. Jos organisaatio tuottaa hyödykkeitä,

on organisaatiolla yleensä eroteltavissa erillinen tuotanto-osasto. Suuremmissa organisaatioissa voidaan omina osastoinaan pitää myös ostoa ja myyntiä, joskin pienemmissä organisaatioissa nämä on yhdistetty usein hallintoon. Koska edellä kuvattujen osastojen työntekijöillä voi olla erilaiset tehtäväkuvat, on tietojärjestelmistä muodostettu eri osastoille omanlaisensa. Tämä ei kuitenkaan estä osastoja käyttämästä toisille osastoille suunnattuja tietojärjestelmiä.

Yleensä tietojärjestelmät on rakennettu kattamaan organisaation kaikkien osastojen tarpeet. Silloin henkilöstö käyttää samaa järjestelmää niin, että järjestelmästä on jokaista osastoa varten oma räätälöity näkymä. Kuva 4.9 esittää, miten tietojärjestelmät jakautuvat yrityksen eri osastojen välillä.



Kuva 4.9 Tietojärjestelmän käyttö yrityksen sisällä (Kettunen 2002).

Myyntiä tukeva tietojärjestelmä on nimeltään CRM, Customer Relationship Management. Asiakkuudenhallintajärjestelmällä hallinnoidaan organisaation asiakkaisiin liittyviä myynti- ja seurantatietoja, tietoja kilpailijoista ja mahdollisista uusista asiakkaista. CRM-järjestelmän avulla pystytään organisaation myyntiä ja myynnin tehokkuutta seuraamaan paremmin niin yritys-, kuin yksilötasolla.

Tuotannossa on yleensä käytössä jokin ERP-tuotannonohjausjärjestelmä. ERP on lyhenne englanninkielen sanoista Enterprise Resource Planning. Lisäksi tuotannossa saatetaan hyödyntää tuotetietojen hallintaan liittyviä PDM-järjestelmiä ja tuotannon ohjaukseen liittyviä MES-järjestelmiä. PDM on lyhenne sanoista Product Data Management ja MES sanoista Manufacturing Execution System. Näillä järjestelmillä hallinnoidaan tuotteen elinkaaren vaiheita raaka-aineista valmiiseen tuotteeseen saakka.

Tuotannonohjausjärjestelmään voidaan määrittää tuotantovaiheiden vaatimat tarpeet työvoimasta, työajasta ja raaka-aineista. Näin koko tuotantoprosessin seuranta on keskitettyä ja tehostaminen helppoa. Tuotantoprosessin virheet ja pullonkaulat huomataan nopeammin.

Keskitetyssä tuotannonohjausjärjestelmässä on tärkeää, että tuotteisiin liittyvää tietoa käsitellään hallitusti. Mikäli tuotannossa tehdään muutoksia tuotekonfiguraatiohin, on esimerkiksi myynnin tiedettävä mitä muutoksia tuotteeseen on tullut. Tuotetiedon hallintajärjestelmän merkitys korostuu tuotteiden määrän kasvaessa.

Ostotoiminnoissa tarvittavia tietoja voidaan hallita ostoreskontraohjelmilla tai toimisto-ohjelmilla. Myynnistä voi löytyä vastaavasti myyntireskontra. Tuntikirjanpitoa varten on olemassa omat ohjelmansa. Organisaation hallinnossa voi olla käytössä useita erilaisia tietojärjestelmiä, joilla hallitaan kirjanpitoa, laskutusta tai viestintää. (Kettunen 2002). Näihin kaikkiin järjestelmiin on koottu oman toimialueensa kannalta keskeiset elementit.

Tietojärjestelmien suuri määrä organisaatiossa johtaa usein siihen, että järjestelmien hallinnoima tieto on tietojärjestelmissä päällekkäistä. Tähän on ratkaisuna tietojärjestelmien välinen integrointi. Yhdistämällä tietojärjestelmiä toisiinsa pystytään organisaation prosesseja automatisoimaan ja tehostamaan. Esimerkiksi myyntiosaston myytyä tuotteen muodostuu siitä tuotanto-osastolle tilaus. Tuotanto-osastolta muodostuu raaka-ainetilaus ostolle ja osto-osastolta ostokirjaus suoraan alihankkijan myyntijärjestelmään. Koko prosessi etenee parhaassa tapauksessa täysin automatisoidusti, mikä on tehokkaampaa verrattuna perinteiseen toimintatapaan, missä työntekijät itse hoitavat prosessin etenemisen soittamalla tai sähköpostitse.

4.2.1. ERP-järjestelmä

Tietojärjestelmien integrointitarpeiden kasvaessa on siirrytty kokonaisvaltaisempiin, koko organisaation kattaviin tietojärjestelmiin. Tietojärjestelmiä voidaan räätälöidä organisaation tarpeiden mukaisesti. Tietojärjestelmä muodostetaan koostamalla yksi yhteinen kokonaisuus organisaation osastojen informaatiotarpeista. Useista moduuleista muodostuneista järjestelmistä käytetään englanninkielistä lyhennettä ERP-järjestelmä, joka juontaa tuotannonohjausjärjestelmien lyhenteestä.

ERP-järjestelmän moduuleja ovat esimerkiksi palkanlaskenta, reskontra, kirjanpito, varastohallinta, projektien hallinta, tuotannonohjaus, materiaalivirtojen hallinta, huolto, resurssien hallinta sekä reklamaatiot. Kun kaikki tieto on saman tietojärjestelmän hallinnoimaa, on tietojen reaaliaikainen tarkastelu organisaatiossa läpinäkyvämpää. Myös organisaation muut yksiköt niin globaalisti kuin paikallisesti saadaan tarvittaessa reaaliajassa hyödyntämään koko organisaation kattavaa tietoa. (Kettunen 2002)

Kun kaikki organisaation sisällä liikkuva tieto on yhden tietojärjestelmän vastuulla, erilaisiin riskeihin varautuminen nousee erittäin tärkeäksi osaksi. Jos tietojärjestelmä jostain syystä kaatuu, voivat seuraukset olla vakavia. Suuri osa organisaation toiminnoista saattaa pysähtyä kun ongelmaa lähdetään korjaamaan. (Kettunen 2002)

4.2.2. Microsoft Dynamics Ax

Microsoft Dynamics Ax on Microsoftin julkaisema toiminnanohjausjärjestelmä. Kilpailevia toiminnanohjausjärjestelmien valmistajia ovat muun muassa SAP ja Oracle. Dynamics Ax on ensisijaisesti suunnattu suurille ja keskisuurille yrityksille. Microsoftin Dynamics tuoteperheen kevyempi versio Dynamics Nav on suunnattu pienyrityksille. Dynamics Ax ei ole käytössä aivan suurimmissa yrityksissä vaan globaalit yritykset käyttävät usein itse tehtyä tietojärjestelmää tai SAPia.

Tässä työssä tutkitussa kehitysprojektissa ei suoranaisesti käytetty Dynamics Ax -järjestelmää, ainoastaan Dynamics Ax -järjestelmän tietokantoja. Dynamics Ax:n tietokanta on hyvin monimutkainen ja siitä löytyy tauluja yli 2000. Taulujen väliset relaatiot on muodostettu usein kahdella tai useammalla kentällä. QlikViewin kannalta tämä aiheuttaa haasteita, koska avaimia joudutaan yhdistämään tai miettimään, mitkä avaimista ovat oleellisia. Kehitysprojektissa toteutetut QlikView-raportit käyttivät lähdeaineistonaan Dynamics Ax -järjestelmän tietokantaa.

5. KETTERÄ OHJELMISTOKEHITYS

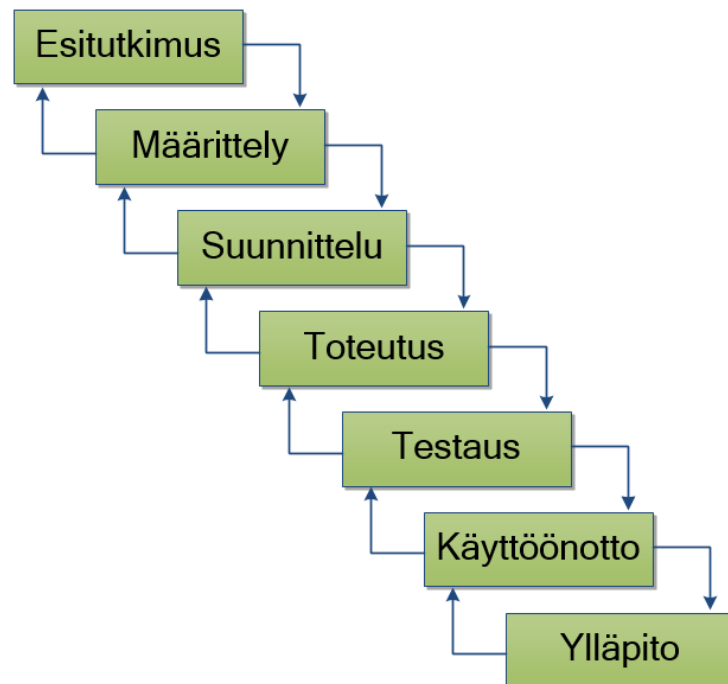
Ketterään ohjelmistokehitykseen voidaan soveltaa erilaisia ohjelmistokehitysmalleja. Nämä eivät kuitenkaan yksin riitä takaamaan ketterää ohjelmistokehitystä, vaan siihen kuuluu olennaisesti myös henkilöresurssit ja niiden huolellinen valinta. Ohjelmistokehitysmalleista käytetään myös nimitystä vaihejakomalli. Vaihejakomalli jakaa ohjelmiston elinkaaren tai kehitystyön eri vaiheisiin. Ohjelmiston elinkaari tarkoittaa aikaa, joka kuluu ohjelmiston kehityksen aloittamisesta ohjelmiston poistamiseen käytöstä (Haikala & Märijärvi 2006).

5.1. Ohjelmistokehitysmalleja

Tässä diplomityössä analysoitu 6pack-projekti ei noudattanut suoraan mitään nykyistä vaihejakomallia. Projekti noudatti malliaan, josta on löydettävissä yhtäläisyyksiä monien tunnettujen vaihejakomallien kanssa.

5.1.1. Lineaarinen vesiputous

Vesiputousmalli on yleinen vaihejakomalli. Siitä löytyy useita eri variaatioita. Yleensä niistä voidaan erotella määrittely-, suunnittelu- ja toteutusvaiheet (Haikala & Märijärvi 2006). Kuvassa 5.1 on esimerkki vesiputousmallista. Mainitut vaiheet löytyvät myös tästä mallista. Niiden lisäksi mallia on täydennetty esitutkimus-, testaus-, käyttöönotto- ja ylläpitovaiheilla. Nuolet eri vaiheiden välillä kuvaavat ohjelmiston elinkaaren kehittymistä ja siirtymistä vaiheesta toiseen.



Kuva 5.1 Vesiputousmalli

Vesiputousmallissa kaikkiin vaiheisiin liittyy laadunvarmistustoimenpiteitä (Haikala & Märijärvi 2006). Näillä toimenpiteillä pyritään havaitsemaan virheet mahdollisimman aikaisin. Mikäli esimerkiksi suunnitteluvaiheessa huomataan jokin ominaisuus tai kokonaisuus, joka on jäänyt kokonaan huomaamatta, palataan määrittelyvaiheeseen ja määritellään kyseinen ominaisuus. Tätä kuvaavat takaisin suuntautuvat nuolet vaiheiden välillä kuvassa 5.1.

Vesiputousmalli etenee vaiheittain seuraavasti:

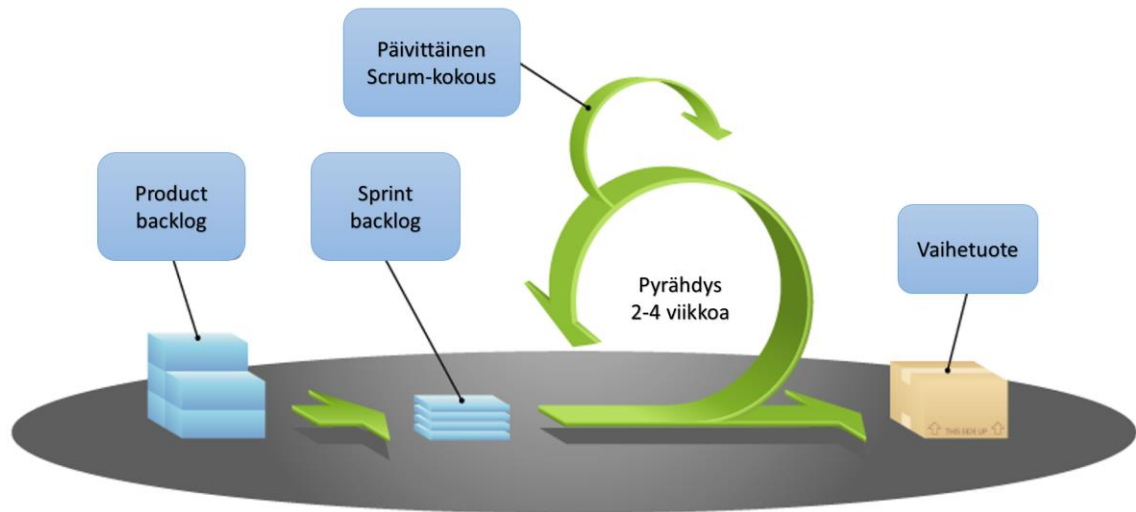
- Esitutkimusvaiheessa kartoitetaan järjestelmän tarpeet joista muodostuvat asiakasvaatimukset. Esitutkimus vastaa kysymykseen, miksi järjestelmä pitäisi tehdä tai miksi sitä ei pitäisi tehdä. Esitutkimus on tärkeä vaihe. Se kannattaa tehdä huolella, sillä huonoista asiakasvaatimuksista ei voi päätyä hyvään järjestelmään. (Haikala & Märijärvi 2006)
- Määrittelyvaiheessa analysoidaan esitutkimuksessa kartoitettuja asiakasvaatimuksia, joista muodostetaan ohjelmistovaatimukset. Nämä ohjelmistovaatimukset määrittelevät toteutettavan järjestelmän. (Haikala & Märijärvi 2006)
- Suunnitteluvaiheessa toteutus kuvataan määrittelyvaiheessa laadituille ohjelmistovaatimuksille. Suunnitteluvaiheessa laaditaan tarkka selonteko, miten ominaisuudet toteutetaan.
- Toteutusvaiheessa toteutetaan itse järjestelmä.
- Testausvaiheessa järjestelmää testataan. Siinä on tarkoituksena löytää virheitä. Testaus voi tapahtua monella tasolla ja testausta helpottamaan on laadittu erilaisia testausmalleja.
- Ohjelmiston testauksen jälkeen, se voidaan saattaa markkinoille ja ohjelmisto siirtyy samalla ylläpitoon. Ylläpidossa ratkaistaan ohjelmiston käytön aikana ilmeneviä ongelmia, korjataan virheitä sekä lisätään mahdollisesti uusia piirteitä. (Haikala & Märijärvi 2006)

5.1.2. Scrum

Scrum on iteratiivinen inkrementaalinen projektinhallintamenetelmä, joka on yleisesti käytössä ketterässä ohjelmistokehityksessä. Scrum-menetelmää kuvattiin ensimmäisen kerran Harvard Business Reviewin artikkelissa ”The New New Product Development Game” vuonna 1986 (Takeuchi & Nonaka 1986). Myöhemmin menetelmää on kuvattu tarkemmin muun muassa Oxford University Press julkaisussa ”The Knowledge-Creating Company: How Japanese Companies Create the Dynamics of Innovation” vuonna 1995 (Nonaka & Takeuchi 1995). Ikujiro Nonaka ja Hirotaka Takeuchi kirjoittivat nämä molemmat julkaisut.

Scrum-menetelmässä ohjelmistoa kehitetään tiiminä. Yhdessä Scrum-tiimissä on keskimäärin seitsemän henkilöä. Scrum-menetelmä koostuu pyrähdyksistä, mitä esittää kuvassa 5.2 alempi ympyränuoli. Pyrähdyksissä työn alla olevaa ohjelmistoa rakennetaan valmiiksi pala kerrallaan. Pyrähdyksiä suoritetaan perä jälkeen ja yhden

pyrähdyksen kesto on yleensä noin kahdesta neljään viikkoa. Kun ohjelmistoa rakennetaan pienissä osissa, on tehtävien seuranta, priorisointi ja suunnittelu helpompaa.



Kuva 5.2 Scrum menetelmä

Pyrähdysten etenemistä arvioidaan ja seurataan päivittäisissä Scrum-kokouksissa, mitä esittää kuvassa 5.2 ylempi ympyränuoli. Scrum-kokoukseen osallistuu koko kehitystiimi. Se kestää noin 15 minuuttia. Kokouksessa kaikki tiimin henkilöt vastaavat kolmeen kysymykseen:

1. Mitä olet tehnyt projektin eteen viimeisen Scrum-kokouksen jälkeen?
2. Mitä olet suunnitellut tekeväsi tänään?
3. Onko sinulla jotain ongelmia, jotka estävät sinua saavuttamasta tavoitteitasi?

Scrum-kokousten tarkoituksena on synkronoida jokaisen tiimihenkilön työpanos, sekä sopia tarvittaessa uusista palaverista, joissa kysymyksistä herännyttä keskustelua voidaan tarkemmin käsitellä.

Kun pyrähdys on saatu päätökseen, tuotteesta voidaan toimittaa asiakkaalle vaihetuote, joka sisältää kaikki siihen mennessä toteutetut toiminnallisuudet. Pyrähdyksen tehtävät arvotetaan jokaisen pyrähdyksen alussa työmääräarvion mukaan, jolloin tehtävien määrää voidaan rajata tarjolla olevien henkilöresurssien mukaisesti. Kesken pyrähdyksen ei työlistalla oleviin tehtäviin tehdä muutoksia.

Scrum-menetelmässä on käytössä kolme erilaista työlistaa (Scrum Alliance 2011):

1. **Product backlog** on tuotteen työlista, joka sisältää kaikki tuotteeseen suunnitellut tehtävät ja vaatimukset.
2. **Release backlog** on julkaisun työlista. Se pitää sisällään kaikki seuraavaan julkaisuun valitut tehtävät.
3. **Sprint backlog** pitää nimensä mukaan sisällään yhdessä pyrähdyksessä toteutettavat tehtävät.

Scrum-menetelmässä on käytössä kolme roolia, joissa henkilöt ovat sitoutuneet projektiin ja jotka vastaavat sen onnistumisesta tai epäonnistumisesta (Scrum Alliance 2011):

1. **Product owner** on tuotteen omistaja ja vastuussa toteutettavan tuotteen ominaisuuksista, liiketoiminnallisesta tuottavuudesta ja töiden priorisoinnista.
2. **Scrum master** vastaa siitä, että työskentelyssä noudatetaan Scrum-prosessia. Hänen vastuullaan on myös poistaa työtä haittaavat esteet ja ulkoiset häiriötekijät, jotta tiimi pystyy työskentelemään tuottavasti.
3. **Team** koostuu usean alueen osaajista. Tiimi valitsee jokaisen sprintin tavoitteet ja päämäärät, perustuen tuotteen omistajan laatimaan priorisoituun työlistaan. Tiimi työskentelee sprinteissä päästäkseen asettamiinsa tavoitteisiin. Tuloksia esitetään tuotteen omistajalle.

5.1.3. Lean

Lean on lyhenne sanoista Lean Thinking. Lean-ajattelun juuret perustuvat Toyotan tuotantjärjestelmään, Toyota Production System, jonka filosofian kehittivät Toyotan insinöörit. Perusajatuksena Lean-menetelmässä on maksimoida asiakkaan kokema lisäarvo minimoiden samalla kaikki turha. (Lean Enterprise Institute 2011)

Lean ohjelmistokehitys käsittää seuraavat seitsemän perusperiaatetta. Jaottelu perustuu kirjaan Lean Software Development: An Agile Toolkit (Poppendieck & Poppendieck 2003):

1. Turhan poistaminen.
2. Oppimisen tehostaminen.
3. Kriittiset päätökset mahdollisimman myöhään.
4. Nopea julkaisu.
5. Toimivaltaa kehitystiimille.
6. Yhtenäinen eheä ohjelmisto.
7. Kokonaisuus ratkaisee.

Turhan poistaminen. Kaikki, mikä ei tuota lisäarvoa loppukäyttäjän näkökulmasta, on turhaa. Loppukäyttäjän näkökulmasta esimerkkejä turhasta ovat tarpeeton toiminnallisuus tai koodi, viivästyminen ohjelmistokehitysprosessissa, epäselvät määrittelyt, byrokratia, sisäisen kommunikoinnin ongelmat ja riittämätön testaus, joka johtaa asioiden toteuttamiseen uudestaan.

Oppimisen tehostaminen. Ohjelmistokehitys on jatkuva oppimisen prosessi. Menestyksellinen työskentely riippuu siitä, kuinka hyvin ohjelmiston kehittäjät ymmärtävät asiakasta, ja kuinka hyvin asiakas ymmärtää todelliset tarpeensa. Kehittäjät saavat palauteprosessilla edellytykset ymmärtää toteutettavan ohjelmiston tarvetta ympäröivässä systeemissä samalla kun asiakas saa käsityksen siitä, mitä vielä tarvitsee kehittää.

Ohjelmiston testaus parantaa oppimista. Kaikki ominaisuudet tulisi testata heti, kun ne on toteutettu. Tarkan dokumentoinnin ja suunnittelutyön sijaan pitäisi rakentaa mieluummin itse tuotetta ja oppia testaamalla sitä.

Kriittiset päätökset mahdollisimman myöhään. Ohjelmistokehitykseen liittyy aina epävarmuutta. Lean-menetelmän näkökulmasta kriittiset päätökset tulisi tehdä vasta sitten, kun on olemassa tosiasioita päätöksenteon tueksi. Jos kriittisiä päätöksiä tehdään olettamuksiin perustuen, voidaan asioita joutua muuttamaan myöhemmin. Mitä monimutkaisempi ohjelmisto on kyseessä, sitä pidemmälle ohjelmistoa pitäisi kehittää, jättäen kriittiset päätökset tekemättä. Kun ohjelmistoa on kehitetty monesta eri suunnasta, on enemmän perusteita tehdä myös kriittisiä päätöksiä.

Nopea julkaisu. Mitä nopeammin ohjelmisto saadaan julkaistua, sitä nopeammin siitä saadaan palautetta seuraavaa iteraatiokierrosta varten. Nopealla julkaisulla varmistetaan, että asiakas saa ohjelmiston, joka vastaa asiakkaan sen hetkisiä tarpeita. Mikäli julkaisu viivästyy, tarpeet saattavat muuttua ja ohjelmisto saattaa olla jo julkaisuhetkellä vanhentunut.

Toimivaltaa kehitystiimille. Tavoitteena on hankkia osaava kehitystiimi, jolla on riittävät toimivaltuudet omaan toimintaansa. Välttämällä mikromanagerointia, johtajan liiallista puuttumista kehittäjien työn yksityiskohtiin, kehitystiimi saadaan keskittymään tuottavaan työhön. Johtajan tehtävänä on kuunnella ohjelmiston kehittäjiä ja tarjota parannusehdotuksia. Kehittäjiä ei tulisi ajatella pelkkänä resurssina. Heitä pitäisi motivoida ja tukea sekä tarjota mahdollisuus asiakkaan kanssa toimimiseen.

Yhtenäinen eheä ohjelmisto. Asiakkaan tulee kokea ohjelmisto alusta loppuun yhtenäisenä kokonaisuutena. Julkaistun ohjelman tulee vastata markkinoitua ohjelmaa. Samoin kustannusten tulee vastata ohjelmaan toteutettuja ominaisuuksia ja käytettävyyttä. Ohjelmisto tulee olla eheä sisäisesti. Ohjelmiston eri komponenttien tulee toimia keskenään tehokkaasti ja joustavasti, mahdollistaen samalla hyvän ylläpidettävyyden. Kehityksen ei pitäisi tapahtua eristyksessä asiakkaasta. Kehittäjien ja asiakkaan välillä informaatiota tulisi vaihtaa jatkuvana virtana, jolloin ongelmakohtiin voidaan tarttua riittävän aikaisessa vaiheessa. Eheä ohjelmisto edellyttää myös yksinkertaistettua koodia. Käännökset ja testaus tulee automatisoida, mikäli se tuottaa ohjelmistolle lisäarvoa.

Kokonaisuus ratkaisee. Virheet kumuloituvat kehitysprosessissa. Kun isoja kokonaisuuksia pilkotaan pienemmiksi ja kehitysvaiheita standardisoidaan, pystytään ongelmien lähteet paikallistamaan ja poistamaan. Mitä monimutkaisempi ohjelmisto, sitä syvemmällä ongelmat usein ovat. Mitä suurempi kehitysprojekti on kyseessä, ja mitä enemmän tahoja sen kehittämiseen osallistuu, sitä keskeisempää on vaalia hyviä alihankkija- ja asiakassuhteita lyhyen tähtäimen kustannusoptimoinnin sijaan. Tällä varmistetaan sujuvasti yhteen nivoutuvien komponenttien muodostama kokonaisuus.

5.2. Henkilöresurssien valinta

Ohjelmistokehitysprojektissa ihmisten ymmärtäminen on keskeisessä roolissa ohjelmistokehitysprojektin alusta lähtien, kun ohjelmistokehitysprojektille laaditaan kehitystiimi. Ohjelmistoprojektin johtajan tulee löytää sopivat ihmiset taitojen, koulutuksen ja kokemuksen perusteella, saada heidät motivoitua ja palkattua. Ohjelmistoprojektin johtajan pitää pystyä vastaamaan oikein seuraaviin kysymyksiin (Peters 2008):

- I. Voivatko nämä henkilöt työskennellä keskenään?
- II. Saavatko nämä henkilöt työn tehtyä?
- III. Pystynkö minä (projektipäällikkö) työskentelemään heidän kanssaan?
- IV. Jos nämä henkilöt työskentelisivät yhdessä, työskentelisivätkö he hyvin yhdessä ryhmänä parhaan osaamisensa mukaan?
- V. Löytyykö näiltä henkilöiltä kaikki tarvittavat taidot?

Mitä tarvitaan tehokkaan ohjelmistokehitysryhmän luomiseen? Määritellessään avaintekijät L. J. Peters on jaotellut vastauksen viiteen eri kokonaisuuteen:

1. Toimialaosaaminen ja -kokemus.
2. Ryhmätyötaidot.
3. Kommunikointitaidot .
4. Joustavuus työtehtävissä.
5. Luonteenpiirteet.

Toimialaosaaminen ja –kokemus. Asiantuntemus vaaditulla toimialalla on keskeisimpiä valintakriteerejä tiimin jäseniä valittaessa. Toimialaosaamista voidaan kartoittaa erilaisin keinoin. Ehdokkaan kyvyistä pääsee hyvin selville esimerkiksi kyselemällä häneltä teknisiä toimialaan liittyviä kysymyksiä.

Ryhmätyötaidot. Taito työskennellä osana tiimiä on ohjelmistokehityksessä keskeistä. Mikäli henkilö ei osaa hyödyntää ympärillään olevaa tiimiä tai kykene auttamaan tiimiä muiden tiimin jäsenten kohtaamissa ongelmissa, menee tiimityöskentelyn idea hukkaan. Ehdokkaan ryhmätyötaitoja voi arvioida selvittämällä ehdokkaan roolit aiemmissa työpaikoissa. Onko ehdokas aiemmissa rooleissa auttanut muita tiimin jäseniä tai koordinoanut töitä tiimin jäsenten kesken yhteisen tavoitteen saavuttamiseksi? Yrittääkö ehdokas ratkaista ongelman loppuun asti itse, vai turvautuuko hän muiden tiimin jäsenten apuun?

Kommunikointitaidot. On tärkeää, että tiimin jäsenet pystyvät ilmaisemaan ajatuksiaan ja mielipiteitään sekä kirjallisesti että suullisesti. Kommunikointitaitoja voidaan selvittää antamalla ehdokkaalle tehtäväksi kirjoittaa lyhyt essee tai pitää pienimuotoinen esitelmä haastattelun aikana. Miten ehdokas ilmaisee itseään paineen alla ja kuinka hän reagoi kysymyksiin?

Joustavuus työtehtävissä. Ohjelmistokehitysprojekteissa kaikki ei mene aina suunnitelmien mukaan. Ongelmiin törmätään väistämättä. Niiden ratkaisu saattaa edellyttää ohjelmistokehittäjälle epämieluisia työtehtäviä, joita ei ehkä ole edes mainittu työ sopimuksissa. Esimerkki epämieluisasta työtehtävästä voisi olla virheiden poistaminen toisen henkilön tekemästä koodista. Kehitysprojektien kannalta on kuitenkin tärkeää, että tiimin jäsenet pystyvät joustamaan työtehtävissä ja toimimaan yhdessä yhteisten tavoitteiden saavuttamiseksi.

Luonteenpiirteet. Ohjelmistokehitysprojektien vetäjillä on harvoin riittävää koulutusta psykologian osa-alueelta arvioidakseen kattavasti ehdokkaan luonteenpiirteen ja henkilökemian soveltuvuutta muuhun tiimiin. Valinta tehdään yleensä sen pohjalta, kenestä projektin vetäjät pitävät tai ketkä ajattelevat asioista samalla tavalla. Tämä sisältää riskejä, sillä kehitystiimiä pitäisi varustaa myös niillä henkilöillä, jotka ajattelevat eri tavoin, jotta mahdollisimman moni näkökulma tulisi projektissa huomioitua.

6. KETTERÄ QLIKVIEW-KEHITYS

Yritysten raportointitarpeet määräytyvät suurimmaksi osaksi yrityksen talousosaston ja johdon esittämien tarpeiden mukaisesti. 6pack-projektia varten tarpeita yleistettiin. Niiden pohjalta koottiin määrittelyt roolipohjaisille QlikView-raporteille. Tärkeänä syynä koko projektin toteuttamiseksi oli Dynamics Ax -järjestelmän vajavaiset raportointiominaisuudet.

6.1. Projektin eteneminen

Projekti toteutettiin käyttämällä ketteristä ohjelmistokehitysmalleista tuttuja menetelmiä. Projektille oli annettu selkeät tavoitteet, joita lähdettiin toteuttamaan seuraavassa kappaleessa esiteltyjen reunaehtojen puitteissa. Projekti eteni pyrähdyksissä, joissa kussakin toteutettiin roolipohjaisia raportteja.

6.1.1. Reunaehdot

6pack-projekti toteutettiin seuraavassa esitettyjä reunaehtoja noudattaen:

- I. Projektissa käytetty uusi järjestelmä QlikView on kehittäjäystävällinen.
 - Järjestelmällä pystyy toteuttamaan raportteja ilman pitkää ohjelmointitautaa. Tämä oli todettu jo ennen projektin alkua yhden henkilön tekemän taustatutkimuksen perusteella.
- II. 6pack-rojektissa toteutettujen raporttien seuraamisella ei ole kriittisiä vaikutuksia.
 - Kenenkään ihmisen terveys ei ollut kiinni raporttien lopputuloksista.
- III. Projektin henkilöstön, Tampereen teknillisen yliopiston opiskelijoiden oli mahdollista toimia projektissa vain kesän ajan.
- IV. Projektissa toteutetut raportit suunniteltiin yrityksen liiketoiminnan tueksi.
 - Raporttien painoarvo ja täsmällisyys on siten virallisia tai julkisia tulosraportteja viitteellisempi.
- V. Yrityksen näkökulmasta on tarpeellista selvittää, kuinka QlikView saadaan myytyä mahdollisimman helposti uusille ja vanhoille Dynamics Ax -asiakkaille.
 - Lisäarvoa tuli saada tuotettua niin osaamisen kasvattamisen kuin liiketoiminnallisen hyödyn muodossa.

6.1.2. Projektin toteutus

Tässä työssä analysoitu kehitysprojekti toteutettiin kesällä 2009. Koko prosessi alkoi jo kesällä 2008, jolloin suoritettiin QlikView-ohjelmistoa koskeva esitutkimus. Esitutkimuksessa arvioitiin QlikView-ohjelmistoa, sen räätälöitävyyttä asiakasympäristöihin, käyttöönoton monimutkaisuutta ja ajallista kestoa sekä QlikViewn tulevaisuuden näkymiä ja markkina-asemaa Suomen markkinoilla. Näiden

pohjalta tehtiin talvella 2009 päätös QlikViewn ottamisesta silloisen SYSteamin, nykyisen EVRYn tuotevalikoimaan.

Tuotevalikoiman laajentamisen yhteydessä tehtiin suunnitelma QlikView osaamisen kartuttamisesta ja valmiiden pakettiraporttien toteuttamisesta nopealla aikataululla. Tämä kehitysprojekti sai nimekseen 6pack. Kehitysprojekti aloitettiin tutkimalla ja kartoittamalla asiakkaita sekä mahdollisia rooleja, joita vastaan kehitysprojektissa toteutettaisiin raportteja. Konsulttien avustuksella laadittiin lista toteutettavista rooleista. Roolit pohjautuivat Dynamics Ax -toiminnanohjausjärjestelmään. Kolme asiakasta ilmoittautui mukaan projektiin.

Toukokuussa 2009 valittiin ensimmäiset kehittäjät mukaan kehitysprojektiin. Kehitysprojekti toteutettiin kesätyöläisvoimin ja projektia varten palkattiin yhteensä viisi SYSteamille täysin uutta työntekijää. Projektiin osallistui myös kaksi jo aiemmin SYSteamilla työskennellyttä. Projektin vetämisestä ja koordinoinnista vastasi kaksi SYSteamin vanhempaa työntekijää.

6.1.3. Roolit

Mukaan ilmoittautuneet asiakkaat saivat valita mieleisensä roolin, joka heille toteutettaisiin. Roolijako on kuvattu taulukossa 6.1.

Taulukko 6.1 6-pack raportit

Rooli	Raportin nimi	Määrittelijä	Pyrähdys
Logistiikkapäällikkö	Vesper	Asiakas	1 / toukokuu
Myyntijohtaja	Camille	Asiakas ja konsultti	1 / toukokuu
Projektipäällikkö	Jinx	Konsultti	2 / kesäkuu
Tuotantopäällikkö	Honey	Konsultti	2 / kesäkuu
Varastopäällikkö	Holly	Konsultti	2 / kesäkuu
Toimitusketjunhallinta	Domino	Asiakas	2,5 / kesäkuu
IT johtaja	Teresa	Konsultti	3 / heinäkuu
Talousjohtaja	Tiffany	Konsultti	3 / heinäkuu

Logistiikkapäällikön, myyntijohtajan ja toimitusketjunhallinnan roolien määrittelyssä oli mukana asiakkaita. Konsultit vastasivat muiden roolien määrittelyistä.

Kutakin roolia määrittelemään valittu konsultti valittiin osaamisensa perusteella. Dynamics Ax:n talous-roolikeskuksen osaja auttoi määrittelemään talousjohtajan roolia, myynti-roolikeskuksen osaja myyntijohtajan roolia jne. Toteutettujen määrittelyjen taso vaihteli suuresti. Kaikista rooleista ei saatu kovinkaan kattavia määrittelyjä. Niitä jouduttiin tarkentamaan projektin edetessä.

6.1.4. Koulutukset

Ensimmäisten projektityöntekijöiden koulutus aloitettiin 12.5.2009. Koulutukseen osallistui kolme työntekijää ja kaksi kouluttajaa. Koulutus aloitettiin tutustuttamalla työntekijät Dynamics Ax -järjestelmään. Yhdellä koulutettavalla oli aiempaa kokemusta Dynamics Ax -toiminnanohjausjärjestelmästä. Kyseinen henkilö oli työskennellyt aiemminkin SYSteamilla Dynamics Ax -kehityksen parissa. Dynamics Ax -koulutus sisälsi lyhyen kahden tunnin perehdyttämisen järjestelmään ja sen toimintoihin sekä kahden tunnin teknisen tuotekatsauksen. Teknisessä katsauksessa tutustuttiin siihen, miten Dynamics AX on rakennettu tietokantojen päälle ja millainen on varsinainen tietokantarakenne. Koulutuksen tavoitteena oli antaa koulutettaville käsitys, miten Dynamics Ax -järjestelmässä näkyviä tietoja voidaan jäljittää tietokantatasolle, ja miten tietokannasta poimitaan vastaava tieto SQL-tietokannan kyselykielen hauilla.

Seuraavana päivänä pidettiin QlikView-peruskoulutus. Yhdellä koulutettavista oli hieman aiempaa kokemusta QlikView-ohjelmistosta aiemmasta työpaikastaan. Koulutuksessa esiteltiin ensin Qlik yrityksenä, ja tämän jälkeen asennettiin ohjatusti QlikView-ohjelma koulutettavien koneille. Koulutus eteni perustoiminnallisuuksien ja QlikViewn peruskäytön opiskelulla. Iltapäivällä käytiin läpi raporttien ulkoasun muokkausmahdollisuuksia. Koulutus oli hyvin pintapuolinen, se ei kattanut läheskään kaikkea QlikViewn toiminnallisuuksista. Tavoitteena oli antaa koulutettaville käsitys, mitä QlikView-ohjelmalla voi tehdä ja miten QlikView-raporttia ja sen ominaisuuksia käytetään.

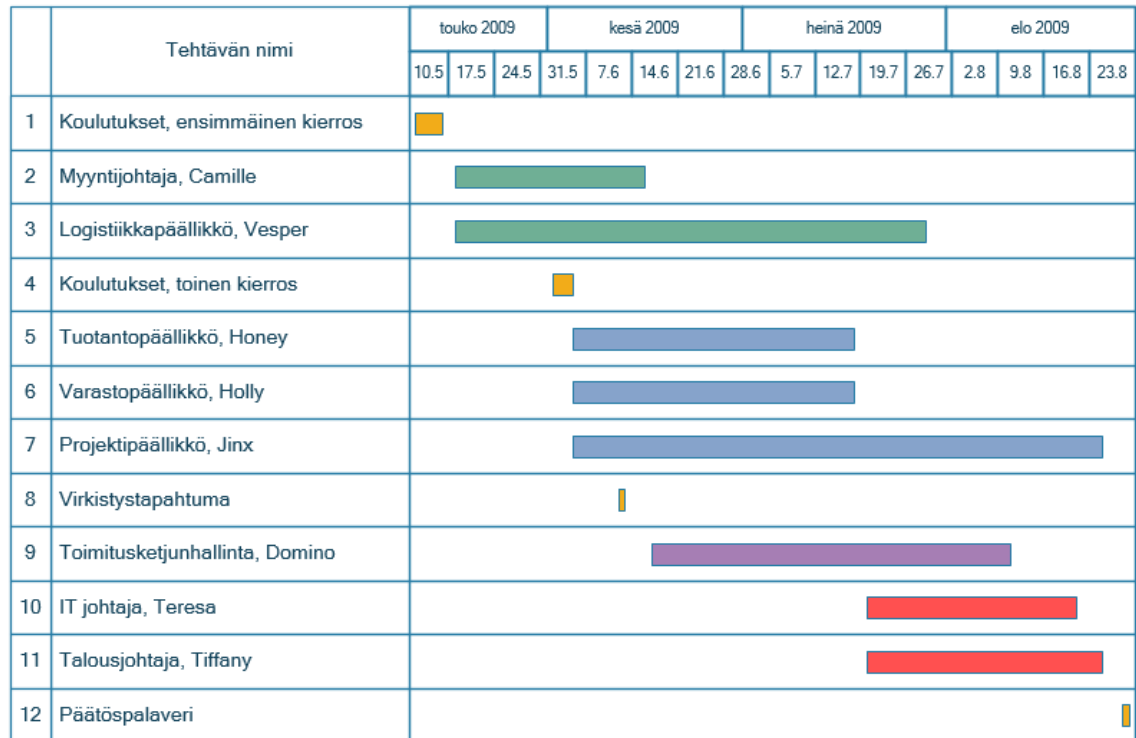
Kolmantena koulutuspäivänä aloitettiin QlikView-kehittäjäkoulutus. Koulutuksen tavoitteena oli opettaa koulutettaville työkalujen käyttö QlikView-perusraporttien kehittämistä varten. Koulutus sisälsi paljon harjoituksia. Teoriaa käytiin läpi harjoitusten lomassa lähinnä itsenäisesti.

Koulutusta jatkettiin vielä seuraavana päivänä koko koulutuksen aihealuetta kattavalla harjoituksella. Tässä harjoituksessa koulutettavat saivat tehtäväkseen rakentaa esimerkkiraportin itsenäisesti alusta loppuun saakka. Tuloksia vertailtiin referenssiraporttiin. Aikaa raportin tekemiseen oli käytettävissä noin neljä tuntia.

Viimeisenä koulutuspäivänä pidettiin myös QlikView Server - ja QlikView Publisher -esittelyt. Siinä esiteltiin komponenttien toiminnallisuudet. Lisäksi käytiin läpi kuinka QlikView-raportti siirretään QlikView Serverin hallintaan.

6.2. Pyrähdykset

Pyrähdykset käynnistyivät koulutusten jälkeen. Niitä oli kaikkiaan neljä. Kuva 6.1 esittää pyrähdysten ajoittumisen kesälle 2009 ja pyrähdysten kestot.



Kuva 6.1 Pyrähdykset

6.2.1. Ensimmäinen pyrähdys

6pack-raportteja suunniteltiin tehtäväksi kahden viikon sykleissä. Tarkoituksena oli saada ensimmäisellä viikolla aikaan toimiva datamalli. Toisella viikolla oli tarkoitus toteuttaa raportin ulkoasu. Datamallin toteutus osoittautui huomattavasti odotettua haasteellisemmaksi ja yhden raportin valmistus kesti keskimäärin noin kuukauden. Tästä ajasta suurin osa kului datamallin rakentamiseen.

Ensimmäinen pyrähdys käynnistyi 18.5.2009. Projektiin osallistui pyrähdyksessä kaksi työntekijää, ja molemmat saivat tehtäväkseen toteuttaa omat raportit. Toteutettaviksi raporteiksi valittiin myyntijohtajan raportti sekä logistiikkapäällikön raportti, koska näiden arveltiin olevan kysytyimpiä.

Ensimmäisenä päivänä tarkastettiin raporttien määrittelyt. Tähän osallistuivat 6pack-projektin vetäjät sekä raporttien toteuttajat. Asiakkaan edustajia tai määrittelyn laadinnassa mukana olleita muita SYSteamin konsultteja ei tässä vaiheessa vielä ollut mukana.

Toinen päivä oli varattu datamallin kehitykselle. Kolmannen päivän iltapäivällä pidettiin datamallin ensimmäinen katselmus. Datamallit olivat puutteellisia, mikä johtui kehittäjien kokemattomuudesta. Kun ensimmäiselle kehitysviikolle osui helatorstai, kehittäjille jäi vain kolme päivää datamallin kehitykselle.

Toisella viikolla oli tarkoitus viimeistellä datamalli ja tehdä ulkoasu siihen kuntoon, että 2-3 suorituskymittarin analysointi onnistuisi raportin avulla. Todellisuudessa ulkoasua saatiin toisella viikolla tehtyä vain vähän ja aika kului pääasiassa datamallien rakenteluun. Samalla viikolla projektitiimi huomasi, että ulkoasun teema oli jäänyt määrittelemättä. Havaittiin, että teema olisi syytä laatia ennen raporttien ulkoasun täsmällisempää toteutusta. Teema laadittiin nopeasti ilman mainostoimiston apua, vaikka suunnitelman mukaan mainostoimistoa piti hyödyntää väripaletin suunnittelussa.

Toisen viikon viimeisenä työpäivänä pidettiin pyrähdysen päätöspalaveri, jossa esiteltiin ja arvioitiin työn tuloksia. Oli selvinnyt, että raporttien kehitys tulisi vaatimaan lisää aikaa. Sen vuoksi päätettiin valmiiden sovellusten esittelyn ajankohdan siirtämisestä viikolla eteenpäin.

Kolmannella toteutusviikolla oli tarkoituksena tehdä raporttiin lisää uusia KPI-näkymiä ja What-If -näköä. Viikko kului kuitenkin datamallin korjauksiin ja aiempien KPI-näkymien kehityksen jatkamiseen. Viikon päätteeksi valmistuivat ensimmäiset versiot molemmista raporteista. Ensimmäinen pyrähdys saatiin näin tehtyä.

Kehityksen oli määrä jatkua myyntijohtajan ja logistiikkapäällikön raporttien kohdalla iteratiivisesti. Mikäli raportteihin saataisiin määriteltä uusia KPI-näkymiä, ne oli tarkoitus toteuttaa 1-2 viikon pyrähdysinä. Molempien raporttien kehitys jatkui What-If -näkömön toteutuksella ja raporttien monikielisyyden toteuttamisella.

6.2.2. Toinen pyrähdys

Toisessa pyrähdyksessä 6pack-projektissa toteutettiin kolme raporttia, varastopäällikön, tuotantopäällikön ja projektipäällikön raportit. Myyntijohtajan ja logistiikkapäällikön raporttien kehitys jatkui samanaikaisesti, joten uusien raporttien kehittäjiksi valittiin kolme uutta työntekijää. Kenelläkään heistä ei ollut aiempaa kokemusta QlikView-ohjelmistosta. Yhdellä työntekijöistä oli yhden TTYn kurssin verran kokemusta Microsoft Dynamics AX -järjestelmästä, kahdelle muulle se oli täysin tuntematon.

Toinen pyrähdys alkoi kuten ensimmäinen pyrähdyskin, uusien työntekijöiden kouluttamisella. Muutamia asioita opittiin ensimmäisen koulutuskierroksen palautteesta ja niitä pyrittiin kehittämään toisen pyrähdysen koulutuksissa.

Toisen pyrähdysen ensimmäinen koulutuspäivä pidettiin 2.6.2009, ja aiheena oli Dynamics AX -perehdytys. Koulutuksessa pyrittiin keskittymään aiempaa enemmän

tiedon hakuun liittyviin asioihin ja tietojen jäljittämiseen Dynamics Ax -tietokannassa. Nämä olivat ensimmäisessä pyrähdyksessä havaittu vaikeiksi asioiksi heti varsinaisen kehityksen lähdettyä käyntiin, joten niihin haluttiin paneutua mahdollisimman aikaisessa vaiheessa.

QlikView-peruskoulutus pidettiin 3.6.2009. Toisen pyrähdyn peruskoulutus ei eronnut sisällöltään mainittavasti ensimmäisen pyrähdyn peruskoulutuksesta. Ainoana eroavaisuutena lienee kouluttajien oman ammattitaidon karttuminen ja sitä kautta koulutuksen ammattimaisempi läpikäynti. Sisällöltään koulutus noudatteli ensimmäisen pyrähdyn peruskoulutuksen vastaavaa sisältöä.

Viimeinen 6pack-projektin koulutus pidettiin 4.6.2009. Aiheena oli QlikView-kehittäjäkoulutus. Koulutuksessa huomioitiin aiemmista koulutuksista saatu palaute. Koulutusta osattiin näin muokata vastaamaan paremmin sille asetettuja vaatimuksia. Koulutuksessa opiskeltiin palautteen perusteella muutamia vaikeaksi miellettyjä kokonaisuuksia, kuten joustavan kalenterikomponentin toteuttaminen ja monikielisyyden tuki. Myöhemmin ilmeni, että vaikka ongelmakohtiin osattiin koulutuksessa paneutua aiempaa paremmin, oli jokaisen kehittäjän kanssa käytävä erikseen kyseiset ongelmakohdat läpi, niiden tultua esiin kehittäjien omissa raporteissa.

Toisen pyrähdyn QlikView-raporttien kehitys aloitettiin 5.6.2009. Seuraava viikko käytettiin kokonaisuudessaan datamallien rakenteluun. Perjantaina 12.6.2009 järjestettiin 6pack-projektin osallistujille Kick-Off virkistystapahtuma, tarkoituksena yhdistää uusia ja vanhoja SYStamiläisiä, ja madaltaa kynnystä uusille työntekijöille lähestyä SYSteamin vanhoja työntekijöitä vastaantulevissa kysymyksissä. Virkistystapahtuma osoittautui menestyksekkääksi ja tilaisuus täytti sille asetetut tavoitteensa.

Datamallien kehitystä jatkettiin seuraavalla juhannusviikolla. Juhannuksen jälkeisellä viikolla suuri osa 6pack-projektin vetäjistä oli lomalla. Raporttikehitystä oli määrä jatkaa itsenäisesti, turvautuen tarvittaessa muihin projektiryhmäläisiin. Palaveriajat oli asetettu valmiiksi kalenteriin, mutta projektin vetäjien puuttuessa palaverien hyötyä ja tarpeellisuutta ei ymmärretty.

6.2.3. Pyrähdys 2,5

Kun myyntijohtajan roolikeskusraportti valmistui juhannusviikolla 2009, raportista vastannut projektiryhmäläinen aloitti uuden toimitusketjunhallinta-roolikeskusraportin kehityksen. Raporttia varten saatiin asiakkaalta lyhyt muutaman rivin määrittely.

Pyrähdysten lisääntyessä raporttien vaiheiden seuraaminen hankaloitui, ja aikatauluttaminen muodostui yhä raporttikohtaisemmaksi. Pyrähdysä oli käynnissä juhannusviikolla kolme, vaikka suunnitelman mukaan toisen pyrähdyn alkaessa

aiemmat raportit piti olla valmiina. Koulutuksia ei enää tässä pyrähdyksessä tarvinnut pitää, mikä nopeutti raporttikehitystä viikolla.

6.2.4. Kolmas pyrähdys

Heinäkuussa 2009 saatiin sekä varastopäällikön että tuotantopäällikön roolikeskusraportit kehitettyä valmiiksi. Logistiikkapäällikön, projektipäällikön ja toimitusketjunhallinnan roolikeskusraportit olivat tässä vaiheessa vielä kesken.

Kolmas pyrähdys aloitettiin 20.7.2009. Siinä toteutettiin IT-johtajan ja Talousjohtajan roolikeskusraportit. Pyrähdys eteni verkkaisesti ja 24.8.2009 saatiin projektin kaikki roolikeskusraportit valmiiksi. Projektin päätöspalaveri pidettiin 28.8.2009.

7. RATKAISUN ARVIOINTI

Projektin onnistumista ja siitä saatuja hyötyjä arvioitiin seuraavilla arviointikriteereillä:

- Osaamisen kasvu,
- asiakastyytyväisyys,
- raporttien laatu,
- ohjelmiston tehokkuus kehittäjän näkökulmasta,
- ohjelmiston kehittäjätyytyväisyys,
- kuka saa jatkopaikan.

Osaamisen kasvu valittiin yhdeksi arviointikriteeriksi, sillä se oli yksi projektin päätavoitteista. Asiakastyytyväisyys on keskeinen mittari jatkuvassa palveluliiketoiminnassa, se oli siksi luonnollinen valinta yhdeksi kriteeriksi. Raporttien laatua arvioitiin, sillä se määrittää osittain projektin tuotosten liiketoiminnallisen arvon. Ohjelmiston tehokkuus ja kehittäjätyytyväisyys valittiin arvioitaviksi tekijöiksi koska projektin päättyessä SYSteamilla olisi tarvittaessa ollut mahdollisuus vaihtaa QlikViewn tilalle toinen raportointiohjelmisto, mikäli ohjelmistossa tai sen kehitettävyydessä olisi ollut suuria puutteita. Yksi projektin päätavoitteista oli palkata SYSteamille uusi henkilö raportointiin. Siksi tämän tavoitteen saavuttamista arvioitiin osana koko projektin onnistumista.

Arviointi toteutettiin haastattelemalla kaikki projektiin osallistuneet henkilöt. Seuraavassa on esitetty haastattelujen pohjalta tehdyt johtopäätökset sekä kehitysideat.

7.1. Haastattelut

Projektin edetessä päätettiin, että prosessi analysoidaisiin. Lisäksi tutkittaisiin onko, prosessista opittavissa jotain. Ennen projektin päättymistä kaikki projektiryhmäläiset haastateltiin henkilökohtaisesti. Haastatteluissa kerättiin kokemuksia ja tunteuksia projektin etenemisestä ja projektissa työskentelemisestä. Haastattelut pidettiin viimeisellä viikolla, jolloin osalla ryhmäläisistä oli vielä projektin työt osittain kesken. Haastattelut kestivät noin puoli tuntia. Haastattelun runko oli vapaamuotoinen. Haastattelun avainkysymykset esitettiin kaikille projektiryhmäläisille, tulosten tulkinta olisi yksiselitteistä ja vertailua olisi mahdollista suorittaa. Haastattelu koettiin pääasiassa positiivisena, se antoi projektiryhmäläisille vapauden ilmaista tunteuksiaan projektista.

7.1.1. Haastattelukysymykset

Projektiryhmäläisten haastatteluissa kysymykset kohdennettiin jokaisen projektiryhmäläisen omaan toimenkuvaan projektissa. Haastatteluissa kysyttiin taulukon 7.1 kysymyksiä.

Taulukko 7.1 Haastattelukysymykset

Miten projekti (koko projekti ja oma osuus) mielestäni onnistui?
Mitä mieltä olet tämän tyyppisestä kehitysprosessista?
Mikä oli helppoa?
Mitä uutta opit?
Mikä oli vaikeaa?
Missä onnistuttiin?
Missä epäonnistuttiin?
Mitä olisi voinut tehdä toisin?
Loman vaikutus/merkitys?
Kokemukset/ajatukset muista konsulteista?
Asiakkaan mukana olon / puuttumisen merkitys?
Puutteet / lisättävää?
Alku, koko projekti ja tuki?
Ohjaus, palaverit ja käytännöt?
Tiesitkö missä mennään projektin edistyessä?
Projektin pituus?
Asetelma?
Ryhmädynamiikka?
Tavoitteet?
Tulosten/tuotteiden arviointi?
Huomioita seuraavaa projektia varten?
Mitä ongelmia asiakkailla oli?
Muuta?

7.2. Arviointi

Projektin onnistumista arvioitiin haastatteluista kerättyjen tulosten pohjalta. Haastateltavat antoivat vastauksia sekä sanallisesti että arvosanoin. Yhteenvedot tehtiin vastauksista.

7.2.1. Projektin kehittäjätyytyväisyys

Projekti sai kehittäjiltä pääasiassa hyvää palautetta, se näkyi myös projektille annetuissa kouluarvosanoissa (1..5). Projektin saamat arvosanat on listattu taulukossa 7.2. Yleisesti projektia pidettiin uskaliaana ja ennakkoluulottomana, mutta silti toimivana menetelmänä, jolla sekä raporteja että osaamista saataisiin hankittua.

Taulukko 7.2 Projektista annetut arvosanat

Arvosana koko projektille	Arvosanat kehittäjän omille tuotoksille
4	4
3	3
4-	3 ½
4	4
4	4
5	4
Ka 4	Ka 4-

Kehittäjien näkökulmasta projekti oli sopivan mittainen. Kaikki projektia varten palkatut kehittäjät olivat TTYn opiskelijoita. Heidän opintonsa olivat vielä kesken. Koulun kesälomaa pidempään projektiin ei suurin osa kehittäjistä olisi pystynyt sitoutumaan.

Projektin työkalut olivat helppokäyttöisiä. Tosin muutamat olivat sitä mieltä, että projektin alussa olisi voinut olla työkalujen koulutusta enemmän. Erityisesti QlikView-ohjelmisto koettiin kehittäjäystävälliseksi. Suurimmat haasteet koettiin datamallin muodostamisessa Dynamics AX pohjalta.

Kaikki kehittäjät olivat yhtä mieltä, että projektissa oppi paljon uusia asioita. Opittuja asioita olivat mm. ohjelmointi, tietokantaosaaminen, tietojärjestelmäosaaminen ja projektityöskentely. Projektin aikana oppi ymmärtämään, millaista on toimia ohjelmistokehittäjänä osana projektia, ja kuinka haastavaa asiakkaan toiveiden ymmärtäminen voi olla.

Kehittäjien mielestä projektissa toteutettujen raporttien määrittelyissä oli selvästi parannettavaa. Osa määrittelyistä muuttui projektin edetessä, eikä jokaista raporttia varten ollut lainkaan täsmällistä määrittelyä. Muutamassa tapauksessa määrittelystä vastaavaa henkilöä oli vaikea tavoittaa projektin edetessä. Tärkein syy tähän oli ajankohta, suurin osa määrittelijöistä oli lomalla projektin aikana.

Kehittäjien mielestä tavoitteita ei oltu esitetty riittävän tarkasti. Lopputulosta ei osattu hahmottaa, koska tavoitteet olivat epäselviä. Myös asiakkaan sitouttaminen projektiin koettiin haasteena. Asiakkaita oli paikoin myös vaikea tavoittaa kesälomien vuoksi.

Näkemykset projektin ohjauksesta ja palavereista olivat kehittäjillä osittain ristiriitaiset. Osa kehittäjistä oli sitä mieltä, että palavereita oli riittävästi ja ohjausta sai sitä pyydettyäessä. Osa totesi, että apua ja ohjausta oli välillä vaikea saada ja varsinkin projektin loppuvaiheessa määräraajojen ja ohjauksen puuttuminen näkyi projektin rönnsyilynä ja tehottomuutena.

Ryhmädynamiikasta kaikki projektiryhmäläiset antoivat hyvää palautetta. Konflikteja ei syntynyt ja kaikki ryhmäläiset tulivat toistensa kanssa hyvin toimeen. Muutama oli sitä mieltä, että ”*tasajako ei ole taktiikkaa*”. Heidän mielestään työtehtäviä olisi pitänyt profiloida kehittäjien osaamisen mukaan. Projektin loppuvaiheessa yksi kehittäjistä keskittyi raporttien kielen kääntämiseen ja tämä sai kehittäjiltä hyvää palautetta. Tällaista profilointia olisi kaivattu jo projektin alkuvaiheessa. Raporttien ulkoasun suunnittelua varten olisi kehittäjien mielestä pitänyt palkata esimerkiksi graafinen suunnittelija tai ulkoistaa koko ulkoasun suunnittelu mainostoimistolle.

Dokumentointi ja sen määrä jakoi mielipiteitä kehittäjien keskuudessa. Osa kehittäjistä oli sitä mieltä, että dokumentointia tehtiin liikaa, jolloin kehitykselle jäi vähemmän aikaa.

Raporteille ei suoritettu lainkaan kokonaisvaltaista lopputestausta, raporttien osat testattiin niitä toteutettaessa. Yksi kehittäjistä mainitsi loppuhaastattelussa että lopputestaus olisi pitänyt suorittaa.

Raporttien jatkokehittäjiltä raportit saivat kriittistä palautetta. Moni asia olisi jatkokehitystä varten pitänyt tehdä toisin tai jättää kokonaan tekemättä. Raporttien räätälöiminen uuden asiakkaan tarpeisiin koettiin asetetuista tavoitteista huolimatta erityisen haastavaksi. Raportit kokivat suuria rakenteellisia muutoksia jatkokehitys- ja räätälöinti vaiheissa. Myös raporttien ulkoasuun jouduttiin tekemään suuria muutoksia, jotka olisi voitu toteuttaa helpommin, mikäli räätälöitävyys olisi huomioitu paremmin ulkoasun suunnitteluvaiheessa.

7.2.2. Raporttien asiakastyytyväisyys

Raportteja toteutettiin kaikkiaan kahdeksan. Kaikkia raportteja ei saatu asiakkaille myytyä. Silti asiakkailta saatu palaute toteutetuista raporteista oli positiivista. Kaikista raporteista on ollut SYSteamille hyötyä. Niitä on käytetty asiakkaille myytyjen uusien raporttien runkoja toteutettaessa.

Negatiivista palautetta ei raporteista saatu, tosin muutoksia ja räätälöintejä asiakkaille toimitettuihin raportteihin toteutettiin paljon. Tätä osattiin odottaa, sillä suurin osa raporteista tehtiin ilman asiakaslähtöistä määrittelyä.

Kokonaan uuden raportin määrittelemisen saattaa olla haastavampaa, kuin olemassa olevan raportin arvioiminen ja sen räätälöiminen. Asiakkaiden mielestä raporteissa oli hyvää monikielisyys sekä valmiin raportin valinnan helppous. Raportit auttoivat asiakkaita ymmärtämään, mitä kaikkea QlikViewillä voidaan tehdä ja minkä tyyppisiä mittauksia liiketoiminta-alueista voidaan ja kannattaa tehdä.

7.2.3. Laatu

Raporttien laatua arvioitiin projektin päättymisen jälkeen useasti. Raporteista on otettu oppia ja moni asia tehtäisiin nyt toisin. Projektin päättymisen aikaan kehittäjät olivat varmoja raporttien laadukkuudesta, mikä näkyi kehittäjien antamissa omien tuotostensa arvioinneissa. Myöhemmin on kuitenkin huomattu, että raporttien käyttöönotto on ollut haasteellista, ja melko suuri osa raporttien määrittämisistä on jouduttu uusimaan.

Asiakkaille toimitettujen raporttien tiedot saatiin poikkeuksetta täsmäämään operatiivisen järjestelmän kanssa. Tästä voidaan päätellä, että vaikka raportteja lähdettiin työstämään vailla minkäänlaista kokemusta raportointityökalusta, tulokseksi on mahdollista saada toimivia raportteja. Tietojen täsmäytyksessä tehtiin räätälöintejä, mutta raporttien logiikka säilyi muuttumattomana.

Raporttien laadun lisäksi projektilla oli myös toinen laadullinen näkökulma. Projektin kehittäjien QlikView-osaaminen koheni projektin seurauksena huomattavasti. Projektilla on ollut suuri merkitys SYSteamin myöhempien QlikView-projektien läpiviemisessä ja näissä projekteissa tuotettujen raporttien toteuttamisessa. Projektia arvioitaessa laadulliset tavoitteet katsottiin toteutuneiksi.

7.2.4. Tehokkuus

Projekti lähti liikkeelle tehokkaasti ja koulutukset saatiin pidettyä suunnitellussa aikataulussa. Sen sijaan raporttien kehittämisen ei voida katsoa sujuneen tehokkaasti. Projektin kehittäjät joutuivat projektin aikana useaan otteeseen odottamaan määrittelyjä asiakkaalta tai SYSteamin määrittelijältä. Kehittäjät joutuivat odottamaan myös Dynamics AX:n liittyvien kysymysten vastauksia. Projektin suunnitteluvaiheessa olisi lomakauden vaikutus pitänyt huomioida projektin toteutuksessa, jotta projekti olisi edennyt tehokkaammin.

Raporttien kehittäminen iteraatioissa katsottiin tehokkaaksi ja erityisen sopivaksi menetelmäksi QlikView-raporttien tapauksessa. QlikViewillä kyettiin toteuttamaan lyhyessä ajassa osa-alue raportista valmiiksi, jonka jälkeen asiakas saattoi testata sitä heti ja antaa palautetta. Nopea asiakastestaus ja palautteen saaminen takasivat, että suurilta virheiltä ja korjaustoilta välttyttiin.

7.2.5. Hyödyt

Kehitysprojekti oli hyödyllinen monessa suhteessa. Liiketoiminnallinen hyöty oli selvä, sillä valmiita raportteja pystyttiin myymään asiakkaille nopealla aikataululla. Raporttien ohessa saatiin myytyä myös räätälöintiin varattua sovitustyötä.

Kehitysprojektista oli hyötyä myös osaamisen kasvamisessa muodossa. SYSteamin henkilöstö sai kokemusta ja osaamista sekä QlikView-raportointityökalusta, että

Dynamics AX -järjestelmästä. Projektissa karttui kehittäjille myös projektiosaamista, toimialaosaamista, konsultointiosaamista, tietokantaosaamista ja ohjelmointiosaamista. Projektin johdolle karttui projektiosaamista.

Kehitysprojektista oli hyötyä myös siksi, että projektissa toteutettuja raportteja voidaan käyttää malleina ja hyödyntää pohjana uusia luotaessa.

7.3. Kehitysideat

Projektin aikana sekä loppuhaastatteluista syntyi muutamia kehitysideoita. Pilotiharjoitus on yksi esiin noussut kehitysidea. Ajatuksena on, että ennen varsinaista projektia pidettäisiin erillinen harjoitus, jossa itse kehitysprojektia testataan. Tarvittavat dokumentit saadaan näin listattua, ja mahdollisiin ongelmakohtiin pystytään varautumaan paremmin.

Dokumentoinnin yhtenäistäminen on toinen kehitysidea. Yhtenäinen dokumenttien tallennuspaikka, jossa pidetään dokumenttien mallipohjia, parantaa projektissa toteutettujen dokumenttien laatua sekä helpottaa dokumentoijien työtä. Mallidokumenttien saatavuus tehostaa myös dokumentointia.

Tiettyjen QlikView-toiminnallisuuksien modularisointi on kolmas kehitysidea. Usein tarvittavia QlikView-toiminnallisuuksia voidaan toteuttaa modulaarisesti eli yleiskäyttöisesti, jolloin niitä voidaan ottaa nopeasti käyttöön uusia raportteja toteutettaessa.

Katselmointi eli raporttien yhteinen läpikäynti on neljäs kehitysidea. Ajatuksena on, että raportteja käydään yksitellen yhdessä läpi ja tutkitaan sitä, miten suurimmat ongelmakohdat on raportissa ratkaistu. Tällä menettelyllä oppimista voidaan syventää niin, että muiden laatimat ratkaisut on omaksuttavissa.

8. YHTEENVETO

6pack-projektissa toteutettiin noin kahden kuukauden aikana kahdeksan erilaista raporttia sekä lukuisia raportteihin liittyviä dokumentteja. Projektin edetessä osallistujat koulutettiin sekä QlikView- että Dynamics AX järjestelmien käyttöön ja kehittämiseen. Raporttien määrittelyjen jakamisen jälkeen toteutukselle oli varattu kutakin raporttia kohden aikaa noin yksi kuukausi. Projektin vetäjät ohjasivat ja rytmittivät projektin etenemistä, sekä auttoivat raporttien toteuttamisessa. Projektin edetessä pidettiin läpikäyntipalavereita, joissa ratkottiin ilmenneitä ongelmia ja esiteltiin valmiita ratkaisuja. Projektin loppuvaiheessa raporteista tehtiin dokumentaatiot. Raporttien valmistuttua kaikki projektin osallistujat haastateltiin ja projektin tuotoksia sekä tuloksia arvioitiin.

Kaikki raportit saatiin tehtyä valmiiksi. Ainoastaan yksi raporteista valmistui selvästi muita myöhemmin, noin viikon myöhässä alkuperäisestä aikataulusta. Toteutetuista raporteista kolme on onnistuttu myymään asiakkaille. Tämä oli selvästi arvioitua heikompi tulos. Muut raportit ovat toimineet pohjana SYSteamin QlikView-kehitysprojekteissa. Useimmista raporteista on katsottu esimerkkiä vastaavaan tarpeeseen liittyvissä asiakasprojekteissa.

Projektin päättyessä arvioitiin raporttien laatua. Raportit täyttivät niille asetetut laadulliset vaatimukset. Jälkikäteen laadussa on kuitenkin huomattu merkittäviä puutteita ja useita ratkaisuja on jouduttu uusimaan. On kuitenkin huomattava, että laadullisten vaatimusten taso on kasvanut kehittäjien kokemuksen ja taitotason kasvun myötä.

Osaamisen kehittämisen näkökulmasta projekti täytti sille asetetut tavoitteet. Työkalujen käyttö- ja ohjelmointiosaamista karttui runsaasti. Projektissa opittiin, että uuden tuotteen omaksuminen on mahdollista parissa kuukaudessa. Lisäksi opittiin, että asiakkaan sitouttaminen projekteihin kesän aikana on erittäin haastavaa, vaikka projektin kustannukset olisivat perinteiseen ohjelmistoprojektiin verrattuna pienemmät.

Projektin seurauksena SYSteamilla – nykyisin EVERY Business Solutions Oyllä – työskentelee kolme projektiin osallistunutta työntekijää. Projektia ei ole lähdetty toteuttamaan uudestaan, koska modulaaristen raporttien toteutusideasta on QlikView-tuotteen tapauksessa toistaiseksi luovuttu. Tähän on syynä se, että kullakin asiakkaalla on omanlaisensa tarpeet, ja kahta samanlaista tapausta ole. Räätelöintejä joudutaan tekemään niin paljon, että on helpompaa toteuttaa alusta lähtien tarpeita vastaava uusi raportti. Muiden järjestelmien omaksumisesta vastaavan projektin muodossa on sitä vastoin suunniteltu.

LÄHTEET

- Aimo, M. 2013. Liiketoimintatiedon hallinta kasvavassa yrityksessä. Kandidaatintyö. Tampere. Tampereen teknillinen yliopisto. Tietotekniikan laitos. 23 p.
- Begg, C. & Connolly, T. 2005. Database Systems: A Practical Approach to Design, Implementation, and Management. 4th edition. Pearson Education, Edinburgh Gate, Harlow. 1374 p.
- Chen, S. J. & Lin, L. 2004. Modeling Team Member Characteristics for the Formation of a Multifunctional Team in Concurrent Engineering. IEEE Transactions on Engineering Management 15, 2, pp. 111–124.
- Cockburn, A. 2006. Agile Software Development: The Cooperative Game. 2nd edition. Addison-Wesley Professional. 504 p.
- Fogel, K. 2005. Producing Open Source Software: How to Run a Successful Free Software Project. 1st edition. O'Reilly Media. 304 p.
- Golfarelli, M. & Rizzi, S. 2009. Data Warehouse Design: Modern Principles and Methodologies. McGraw-Hill/Osborne. 480 p.
- Haikala, I. & Märijärvi, J. 2006. Ohjelmistotuotanto. 11 painos. Jyväskylä, Gummerrus Kirjapaino Oy. 440 p.
- Hannula, M. Tietojohtamisen peruskurssi. 2009. Tampere, Tampereen teknillinen Yliopisto. Luentomoniste.
- Highsmith, J. 2004. Agile Project Management: Creating Innovative Products. Boston, Addison-Wesley. 277 p.
- Jacot, A., Miller, J.E., Jacot, M. & Stern, J.A. 2009. J. D. Edwards EnterpriseOne: The Complete Reference. Oracle Press. 955 p.
- Kettunen, H. & Myllyaho, M. 2007. Agile Software Development Practices in the Software Project Work Education. University of Oulu. 12 p.
- Kettunen, S. 2002. Tietojärjestelmän ostaminen. Porvoo, WSOY. 191 p.

Kimball, R. & Caserta, J. 2004. The Data Warehouse ETL Toolkit. John Wiley & Sons. 528 p.

Larman, C. 2003. Agile and Iterative Development: A Manager's Guide. 1st edition. Addison-Wesley Professional. 368 p.

Lean Enterprise Institute [www]. [Viitattu 23.5.2011]. Saatavissa: <http://www.lean.org/WhatsLean/>

Lueg, C. 2001. Information, Knowledge and Networked Minds. Journal of Knowledge Management 5, 2, pp. 151-160.

Moss, L. T. & Atre, S. 2003. Business Intelligence Roadmap. Boston, Pearson Education. 543 p.

Nonaka, I. & Takeuchi, H. 1995. The knowledge-creating company: How Japanese companies create the dynamics of innovation. New York, Oxford University Press. 304 p.

Peters, L.J. 2008. Getting Results from Software Development Teams. 1st edition. Microsoft Press. 400 p.

Poppendieck, M. & Poppendieck, T. 2003. Lean Software Development: An Agile Toolkit. Addison-Wesley Professional. 240 p.

Qlik [www]. [Viitattu 13.10.2009]. Saatavissa: <http://www.qlik.com/>

Richardson, J. 2009. Business Intelligence Platform Adoption Intentions [www]. [Viitattu 13.10.2009]. Saatavissa: <http://my.gartner.com/portal/server.pt?open=512&objID=260&mode=2&PageID=3460702&resId=1140412&ref=Browse>

Richardson, J., Schlegel, K., Sallam, R. L. & Hostmann, B. 2009. Magic Quadrant for Business Intelligence Platforms [www]. [Viitattu 15.10.2009]. Saatavissa: <http://mediaproducts.gartner.com/reprints/oracle/article56/article56.html>

Schwaber, K. 2004. Agile Project Management with Scrum. Redmond, Microsoft Press. 163 p.

Schwaber, K. 2007. The Enterprise and Scrum. 1st edition. Microsoft Press. 176 p.

Schwalbe, K. 2006. Information Technology Project Management. 5th edition. Boston, Thomson Course Technology. 521 p.

Scrum Alliance [www]. [Viitattu 27.3.2011]. Saatavissa: <http://www.scrumalliance.org>

Stepanek, G. 2005. Software Project Secrets: Why Software Projects Fail. Berkeley, Apress. 192 p.

Takeuchi, H. & Nonaka, I. 1986. The New New Product Development Game. Harvard Business Review. January-February. pp. 285-305.

Tuulos, V. 2008. Agile Project Management with Scrum. Master of Science Thesis. Tampere University of Technology. Tietotekniikan laitos. 56 p.

Virtanen, P. Tietojohtamisen peruskurssi. 2011. Tampere, Tampereen teknillinen Yliopisto. Luentomoniste.

Vuojamo, T. 2008. Development Environment for a Software Project. Master of Science Thesis. Tampere University of Technology. 48 p.